

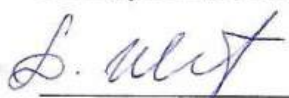
Учреждение образования  
«Белорусский государственный университет культуры и искусств»

Факультет культурологии и социально-культурной деятельности

Кафедра информационных технологий в культуре

СОГЛАСОВАНО

Заведующий кафедрой



Т.С. Жилинская

«24» апреля 2025 г.

СОГЛАСОВАНО

Декан факультета



Н.Е. Шелупенко

«30» мая 2025 г.

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС  
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

**ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В КУЛЬТУРОТВОРЧЕСТВЕ**

*для специальности 6-05-01314-03*

Социально-культурный менеджмент и коммуникации

Составители:

Т.И. Песецкая, доцент кафедры информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств», кандидат физико-математических наук

С.А. Шатько, преподаватель кафедры информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств»

Рассмотрено и утверждено  
на заседании Совет факультета культурологии и социально-культурной деятельности, протокол № 10 от 27.06.2025 г.

## СОСТАВИТЕЛИ:

Т.И. Песецкая, доцент кафедры информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств», кандидат физико-математических наук

С.А. Шатько, преподаватель кафедры информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств»

## РЕЦЕНЗЕНТЫ:

кафедра компьютерных технологий и систем факультета прикладной математики БГУ

С.В. Вабищевич, доцент кафедры информатики и методики преподавания информатики учреждения образования «Белорусский государственный педагогический университет имени Максима Танка», кандидат педагогических наук, доцент

## РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

*кафедрой* информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств» (протокол № 8 от 24.04.2025);

Совет факультета культурологии и социально-культурной деятельности, протокол № 10 от 27.06.2025 г.

## СОДЕРЖАНИЕ

|                                                                       |    |
|-----------------------------------------------------------------------|----|
| 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА .....                                        | 4  |
| 2. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ.....                                          | 7  |
| КОНСПЕКТ ЛЕКЦИЙ.....                                                  | 7  |
| Лекция 1. Понятие “искусственный интеллект” .....                     | 7  |
| Лекция 2. Нейронные сети. Обучение нейронных сетей .....              | 10 |
| Лекция 3. Понятие больших языковых моделей (LLM) .....                | 22 |
| Лекция 4. Генеративный ИИ на основе LLM и его применение .....        | 25 |
| Лекция 5. Промпт-инжиниринг: понятие и основные принципы .....        | 40 |
| Лекция 6. Разработка промптов для сферы культуры.....                 | 59 |
| 3. ПРАКТИЧЕСКИЙ РАЗДЕЛ .....                                          | 63 |
| 3.1 ОПИСАНИЕ ЛАБОРАТОРНЫХ РАБОТ .....                                 | 63 |
| 3.2 ОПИСАНИЕ ПРАКТИЧЕСКИХ РАБОТ .....                                 | 68 |
| 4. РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ.....                                        | 74 |
| 4.1 Задания для контролируемой самостоятельной работы студентов ..... | 74 |
| 4.2 Перечень контрольных вопросов по дисциплине .....                 | 75 |
| 4.3 Перечень вопросов для самоподготовки .....                        | 75 |
| 4.4 Перечень вопросов к зачету .....                                  | 77 |
| 5. ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ.....                                        | 79 |
| 5.1 Учебная программа.....                                            | 79 |
| 5.2 Учебно-методическая карта учебной дисциплины .....                | 81 |
| 5.3 Список основной литературы .....                                  | 82 |
| 5.4 Список дополнительной литературы .....                            | 82 |

## 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная дисциплина «Искусственный интеллект в культуротворчестве» предназначена для студентов специальности 6-05-01314-03 социально-культурный менеджмент и коммуникации, профилизация «Мультимедийные технологии и цифровые коммуникации».

Изучение учебной дисциплины «Искусственный интеллект в культуротворчестве» направлено на обучение студентов теоретическим основам и методам использования искусственного интеллекта для решения задач в сфере культуры.

Целью преподавания учебной дисциплины является приобретение студентами специальности 6-05-01314-03 социально-культурный менеджмент и коммуникации, необходимого уровня компетентности в области использования искусственного интеллекта на основании методологии промпт-инжиниринга для решения различных творческих и исследовательских задач в социокультурной деятельности.

В соответствии с учебным планом учреждения высшего образования по специальности 6-05-0314-03 Социально-культурный менеджмент и коммуникации, профилизации «Мультимедийные технологии и цифровые коммуникации» освоение образовательной программы по учебной дисциплине «Искусственный интеллект в культуротворчестве» должно обеспечивать формирование следующей специальной компетенции:

СК-38. Реализовывать на практике основные функции управления.

СК-39. Выявлять и понимать особенности сотрудничества Республики Беларусь с зарубежными странами и международными организациями

СК.-40. Использовать технологии искусственного интеллекта для разработки социально-культурных проектов

Задачи изучения учебной дисциплины:

- формирование системы базовых знаний о языковых моделях, их архитектуре, принципах обучения и способах взаимодействия с искусственным интеллектом;

- усвоение этических и социальных аспектов, связанных с использованием искусственного интеллекта и языковых моделей в культуротворчестве;

- развитие творческого подхода к поставленной задаче в процессе практической деятельности учащихся;

- изучение принципов, стратегий и методов разработки промптов для решения конкретных задач в сфере культуры;

- освоение практических приемов промпт-инжиниринга для решения задач в социокультурной деятельности.

В результате изучения учебной дисциплины «Искусственный интеллект в культуротворчестве» студенты должны *знать*:

- определение понятия “искусственный интеллект” и его ключевые характеристики;

- принципы работы нейронных сетей;

- этические принципы применения искусственного интеллекта;

- правовые принципы применения искусственного интеллекта;

- основные концепции интеграции промпт-инжиниринга в культурную индустрию;

- примеры применения промпт-инжиниринга в культуре;

- принципы разработки промптов для различных сфер культуры.

Студенты должны *уметь*:

- определять и объяснять ключевые характеристики искусственного интеллекта;

- применять нейронные сети и языковые модели для анализа и создания культурного контента;

- оценивать социальные и культурные последствия автоматизации процессов с помощью искусственного интеллекта;

- интегрировать промпт-инжиниринг в рабочие процессы культурной индустрии;

- создавать промпты для визуального искусства, литературы, музыки и театра;

- разрабатывать промпты для маркетинга и рекламы в сфере культуры;

- создавать ИИ-инструменты для анализа культурного контента и автоматической обработки информации.

Студенты должны *овладеть* следующими навыками:

- применения нейронных сетей и языковых моделей для анализа и генерации культурного контента;

- анализирования и оценивания влияния автоматизации на культуру и общество;

- разработки промптов для различных задач сферы культуры;

– создавать ИИ-инструменты для анализа культурного контента и автоматической обработки информации.

Содержание УМК по данной учебной дисциплины направлено на формирование академических компетенций, включающих знания и умения

по изученным учебным дисциплинам, умение учиться, а также социальных и профессиональных компетенций.

На лекциях особое внимание уделяется рассмотрению теоретических аспектов решения задач сферы культуры с нейронных сетей. Практические и лабораторные занятия направлены на формирование умений практического использования полученных знаний для решения прикладных задач социально-культурной сферы.

Учебным планом на изучение учебной дисциплины «Визуальное программирование» всего предусмотрено 90 часов, из них 44 часа – аудиторные занятия. Примерное распределение аудиторных часов по видам занятий: лекции – 12 часов, практические занятия – 18 часов, лабораторных – 14 часов

Дисциплина рассчитана на один семестр. Текущий контроль осуществляется при выполнении и сдаче отчетов практических и лабораторных работ. Промежуточная форма контроля – устный опрос. Рекомендуемая форма контроля знаний – зачёт.

## 2. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

### КОНСПЕКТ ЛЕКЦИЙ

#### Лекция 1. Понятие “искусственный интеллект”

Программный код, технология или алгоритм, которые позволяют машине имитировать, развивать или демонстрировать человеческое познание или поведение, можно определить как искусственный интеллект (ИИ).

Искусственный интеллект – это очень широкое понятие, в которое входят машинное обучение и глубокое обучение (рис. 1).

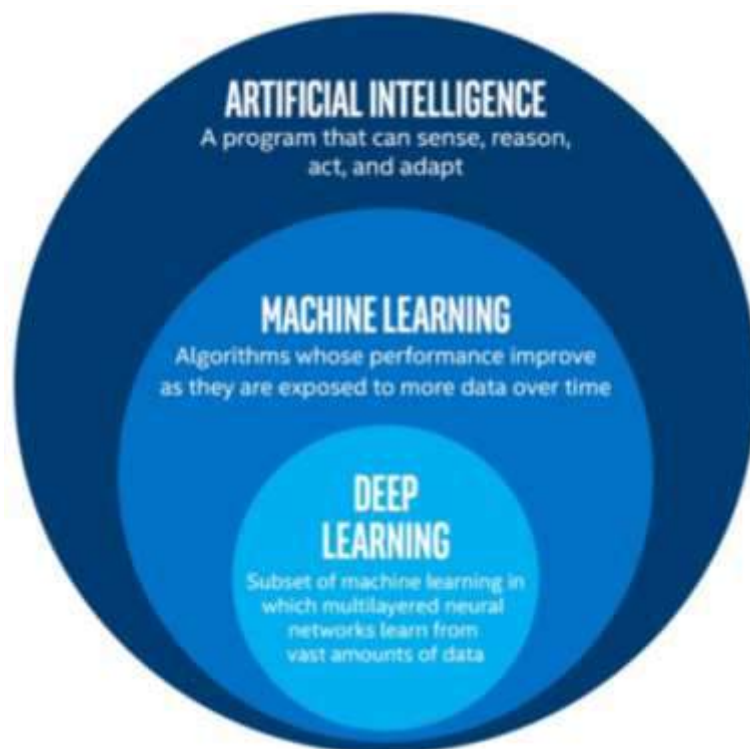


Рис. 1. Структура понятия “искусственный интеллект”

В широком смысле под искусственным интеллектом подразумевается:

- обучение машин действовать и думать, как люди;
- создание программ, которые будут сами действовать без участия человека.

Другое направление ИИ связано с тем, чтобы дать машинам больше когнитивных и сенсорных возможностей. Это направление связано с:

- анализом изображений и видео, с обработкой и пониманием речи и т.д.
- созданием приложений для интеллектуального решения проблем с использованием алгоритмов.

Речь идет о разработке программ или алгоритмов таким образом, чтобы они могли учиться и совершенствоваться с течением времени под воздействием новых данных.

Более узкое направление ИИ – чисто вычислительный подход и подход для оптимизации, в котором производится манипуляция данными таким образом, чтобы получить неочевидные результаты. И здесь ИИ – это инструмент, который используется компьютером для автоматического выполнения задач практически без вмешательства человека.

Искусственный интеллект – это сложная серия алгоритмов, которые что-то делают с поступающей информацией. Мы будем исходить из того, что искусственный интеллект – это набор технологий, который позволяет извлекать знания из данных. Таким образом, это любая система, которая изучает или понимает шаблоны или закономерности в этих данных и может идентифицировать их, а затем воспроизводить их на новой информации.

Будем исходить из того, что искусственный интеллект – это не симулятор человеческого интеллекта, а ИИ – это машинное обучение. И технология машинного обучения – это использование математики в компьютерах для поиска неких шаблонов или закономерностей в данных. Эти данные могут быть структурированными или неструктурированными.

Единственное различие между машинным обучением и технологиями, которые были до этого, состоит в том, что раньше людям приходилось вручную кодировать эти шаблоны. Но компьютеры могут найти эти шаблоны или закономерности самостоятельно, используя математику. Таким образом, для нас искусственный интеллект – это набор математических алгоритмов, которые позволяют компьютерам находить шаблоны, о которых мы, возможно, даже не подозреваем, без необходимости кодировать их вручную.

Разберем понятия *искусственного интеллекта*, *машинного обучения*, *глубокого обучения* и *нейронных сетей* (рис. 2). Эти термины иногда используются взаимозаменяемо, но они не относятся к одному и тому же.

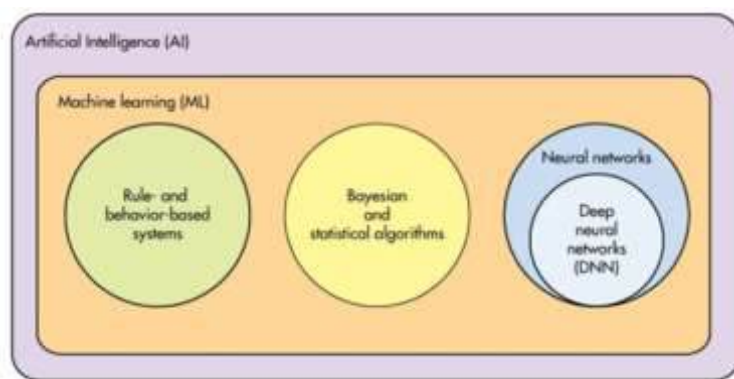


Рис. 2. Соотношение понятий “искусственный интеллект”, “машинное обучение”, “глубокое обучение” и “нейронные сети”

*Искусственный интеллект* – это область информатики, занимающаяся симуляцией интеллектуального поведения. Системы ИИ, как правило, демонстрируют поведение, связанное с человеческим интеллектом, такое как планирование, обучение, рассуждение, решение задач, представление знаний, восприятие, движение и манипуляция, и в меньшей степени социальный интеллект и креативность.

*Машинное обучение* – это подмножество ИИ, которое использует компьютерные алгоритмы для анализа данных и принятия разумных решений на основе того, что они узнали, без явного программирования. Алгоритмы машинного обучения обучаются на больших наборах данных и учатся на примерах. Они не следуют алгоритмам, основанным на правилах. Машинное обучение – это то, что позволяет машинам самостоятельно решать задачи и делать точные прогнозы, используя предоставленные данные.

*Глубокое обучение* – это специализированный раздел машинного обучения, который использует многоуровневые нейронные сети для имитации принятия человеческих решений. Алгоритмы глубокого обучения могут маркировать и классифицировать информацию и идентифицировать шаблоны – закономерности. Это то, что позволяет системам искусственного интеллекта постоянно учиться в процессе работы и повышать качество и точность результатов, определяя правильность принятых решений.

## Лекция 2. Нейронные сети. Обучение нейронных сетей

Идея искусственных *нейронных сетей* основывается на биологических нейронных сетях, хотя они работают совсем по-другому.

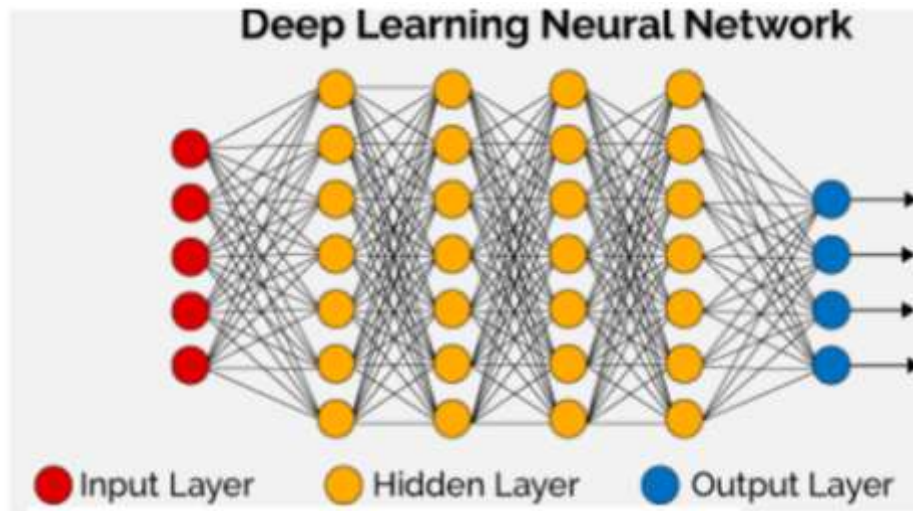


Рис. 3. Глубокое обучение нейросети

Нейронная сеть в ИИ представляет собой набор небольших вычислительных блоков, называемых нейронами, которые принимают входящие данные и учатся принимать решения с течением времени (рис 3).

Нейронные сети часто являются многоуровневыми и становятся более эффективными по мере увеличения объема наборов данных, в отличие от других алгоритмов машинного обучения.

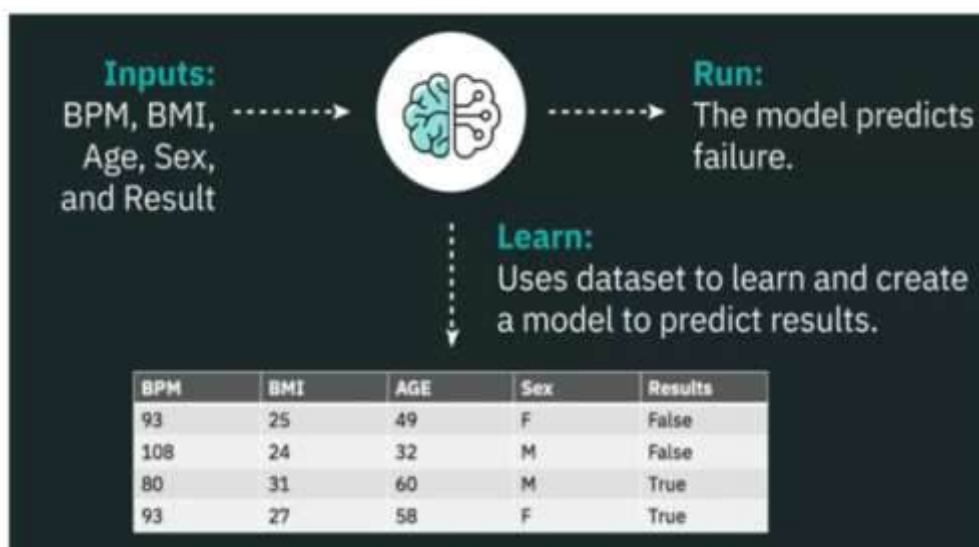


Рис. 4. Пример модели прогнозирования на основе данных

Машинное обучение, подмножество искусственного интеллекта, использует компьютерные алгоритмы для анализа данных и принятия разумных решений на основе того, что алгоритмы изучили. Вместо того, чтобы следовать алгоритмам, основанным на правилах, машинное обучение само строит модели для классификации и прогнозирования на основе данных.

Например, что, если мы хотим определить, может ли возникнуть проблема с нашим сердцем, с помощью машинного обучения (рис. 4)? Можем ли мы это сделать. И ответ – да. Допустим, нам даны такие данные, как количество ударов в минуту, вес тела, возраст и пол. С машинным обучением и этим набором данных, мы можем изучить и создать модель, которая с учетом входных данных будет предсказывать результаты.

Так в чем же разница между этим подходом и просто использованием статистического анализа для создания алгоритма? *Алгоритм* – это математическая техника. В традиционном программировании мы берем данные и правила и используем их для разработки алгоритма, который даст нам ответ.

В этом примере (рис. 4), если бы мы использовали традиционный алгоритм, мы бы взяли данные, такие как сердечный ритм, возраст, вес тела и пол и использовали эти данные для создания алгоритма, который определит, будет ли сердце работать нормально или нет. По сути, это было бы выражение *if – else*. Когда мы отправляем входные данные, мы получаем ответы, основанные на том, какой алгоритм мы определили, и этот алгоритм не изменится от данных.

Машинное обучение, с другой стороны, берет данные и ответы и уже потом само создает алгоритм. Вместо того, чтобы получить ответы в конце, у нас уже есть ответы. А то, что мы получаем здесь, – это набор правил, определяющих модель машинного обучения. И эта модель определяет правила и оператор *if – else* при получении входных данных. В отличие от традиционного алгоритма она может постоянно обучаться и использоваться в будущем для прогнозирования значений.

Машинное обучение опирается на определение правил путем изучения и сравнения больших наборов данных, чтобы найти общие закономерности (рис. 5).

Например, мы можем создать программу машинного обучения с большим объемом изображений птиц, и обучить модель возвращать название птицы всякий раз, когда мы даем изображение птицы. Когда для модели отображается изображение птицы, она маркирует изображение с некоторой степенью достоверности. Этот тип машинного обучения называется контролируемым обучением, где алгоритм обучается на данных, размеченных человеком.



Рис. 5. Машинное обучение

Чем больше примеров мы предоставляем контролируемому алгоритму обучения, тем точнее он производит классификацию новых данных.

Неуправляемое обучение, это еще один тип машинного обучения, которое основывается на предоставлении алгоритму неразмеченных данных и позволяет ему самостоятельно находить шаблоны. Вы предоставляете просто входные данные, и позволяете машине делать выводы и находить шаблоны. Этот тип обучения может быть полезен для кластеризации данных, когда данные группируются в соответствии с тем, насколько они похожи на своих соседей и отличаются от всего остального. Как только данные кластеризованы, можно использовать различные методы для изучения этих данных и поиска шаблонов.

Например, можно создать алгоритм машинного обучения с постоянным потоком сетевого трафика и позволить ему независимо изучать активность в сети – базовый уровень, нормальную сетевую активность, а также выбросы и, возможно, злонамеренное поведение, происходящее в сети.

## Как машины ведут себя при пожаре

| Классическое<br>программирование                                                                  | Машинное<br>обучение                                                                           | Обучение<br>с подкреплением              |
|---------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|------------------------------------------|
| «Я просчитал все варианты<br>событий и ты сейчас<br>должен связать верёвку<br>из хлебного мякиша» | «По статистике люди гибнут<br>в 6% пожаров, поэтому<br>рекомендую вам умереть<br>прямо сейчас» | «Да просто беги от огня<br>AAAAAAAAAAAA» |

Рис. 6. Обучение с подкреплением

Третий тип алгоритма машинного обучения, обучение с подкреплением, это алгоритм машинного обучения с набором правил и ограничений и позволяет ему учиться достигать целей (рис. 6). Вы определяете состояние, желаемую цель, разрешенные действия и ограничения. И алгоритм выясняет, как достичь цели, пробуя различные комбинации разрешенных действий, и его награждают или наказывают в зависимости от того, было ли решение правильным. Алгоритм изо всех сил старается максимизировать свои вознаграждения в рамках предусмотренных ограничений. И вы можете использовать обучение с подкреплением, чтобы научить машину играть в шахматы или преодолеть какие-либо препятствия.

Таким образом, машинное обучение – это широкая область, и мы можем разделить его на три разные категории: контролируемое обучение, неконтролируемое обучение и обучение с подкреплением. И есть много разных задач, которые мы можем решить с помощью них.

В контролируемом обучении, в наборе данных есть метки, и мы используем их для построения модели классификации данных. Это означает, что, когда мы получаем данные, у них есть метки, которые говорят о том, что представляют эти данные. В примере с сердцем, у нас была таблица с метками, это сердечный ритм, возраст, пол и вес. И каждой такой метке соответствовали значения.

При неконтролируемом обучении у нас нет меток, и мы должны обнаружить эти метки в неструктурированных данных. И такие вещи обычно делаются с помощью кластеризации.

Обучение с подкреплением – это другое подмножество машинного обучения, и оно использует вознаграждение для наказания за плохие действия или вознаграждение за хорошие действия.

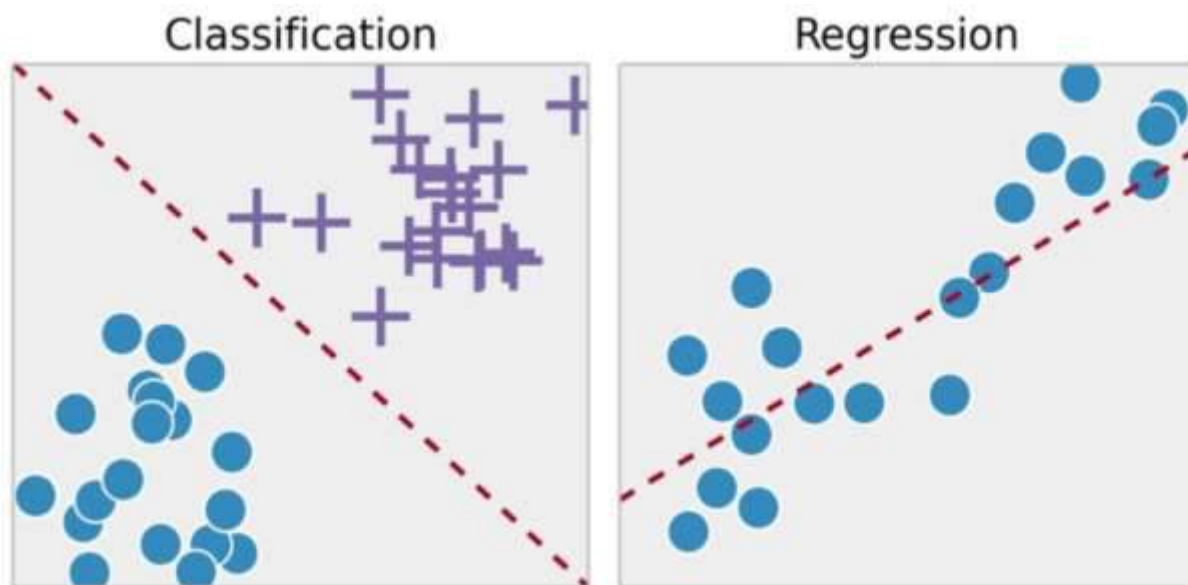


Рис. 7. Категории контролируемого обучения

Мы можем разделить контролируемое обучение на три категории: регрессия, классификация и нейронные сети (рис. 7).

Модели регрессии строятся с учетом взаимосвязей между признаками  $x$  и результатом  $y$ , где  $y$  – непрерывная переменная. По сути, регрессия оценивает непрерывные значения.

Нейронные сети относятся к структурам, которые имитируют структуру человеческого мозга.

Классификация, с другой стороны, фокусируется на дискретных значениях, которые она идентифицирует.

Мы можем назначить дискретные результаты  $y$  на основе многих входных признаков  $x$ . В примере с сердцем, учитывая набор признаков  $x$ , таких как удары в минуту, вес тела, возраст и пол, алгоритм классифицирует выходные данные  $y$  как две категории: истина или ложь, предсказывая, будет ли сердце работать нормально или нет.

В других классификационных моделях мы можем классифицировать результаты по более чем двум категориям.

Например, прогнозирование, является ли данный рецепт рецептом индийского, китайского, японского или тайского блюда. И с помощью классификации мы можем извлечь особенности из данных. Особенности в этом примере сердцем, это сердечный ритм или возраст.

Особенности – это отличительные свойства шаблонов ввода, которые помогают определить категории вывода (рис. 8).

| BPM | BMI | AGE | Sex | Results |
|-----|-----|-----|-----|---------|
| 93  | 25  | 49  | F   | False   |
| 108 | 24  | 32  | M   | False   |
| 80  | 31  | 60  | M   | True    |
| 93  | 27  | 58  | F   | True    |

Рис. 8. Особенности

Здесь каждый столбец является особенностью, а каждая строка – точкой ввода данных.

Классификация – это процесс прогнозирования категории заданных точек данных. И наш классификатор использует обучающие данные, чтобы понять, как входные переменные относятся к этой категории.

Что именно мы подразумеваем под обучением? Обучение подразумевает использование определенного алгоритма обучения для определения и разработки параметров модели. Хотя для этого есть много разных алгоритмов, с точки зрения непрофессионала, если вы тренируете модель, чтобы предсказать, будет ли сердце работать нормально или нет, есть истинные или ложные значения, и вы будете показывать алгоритму некоторые реальные данные, помеченные как истинные, затем снова показывая данные, помеченные как ложные, и вы будете повторять этот процесс с данными, имеющими истинные или ложные значения. И алгоритм будет изменять свои внутренние параметры до тех пор, пока он не научится распознавать данные, которые указывают на то, что есть сердечная недостаточность или ее нет.

При машинном обучении мы обычно берем набор данных и делим его на три набора: наборы обучения, проверки и тестирования. Набор обучения – это данные, используемые для обучения алгоритма. Набор проверки используется для проверки наших результатов и тонкой настройки параметров алгоритмов. Данные тестирования – это данные,

которые модель никогда не видела прежде и которые используются для оценки того, насколько хороша наша модель.

Модель машинного обучения – это алгоритм, используемый для поиска закономерностей в данных без программирования в явном виде (рис. 9).

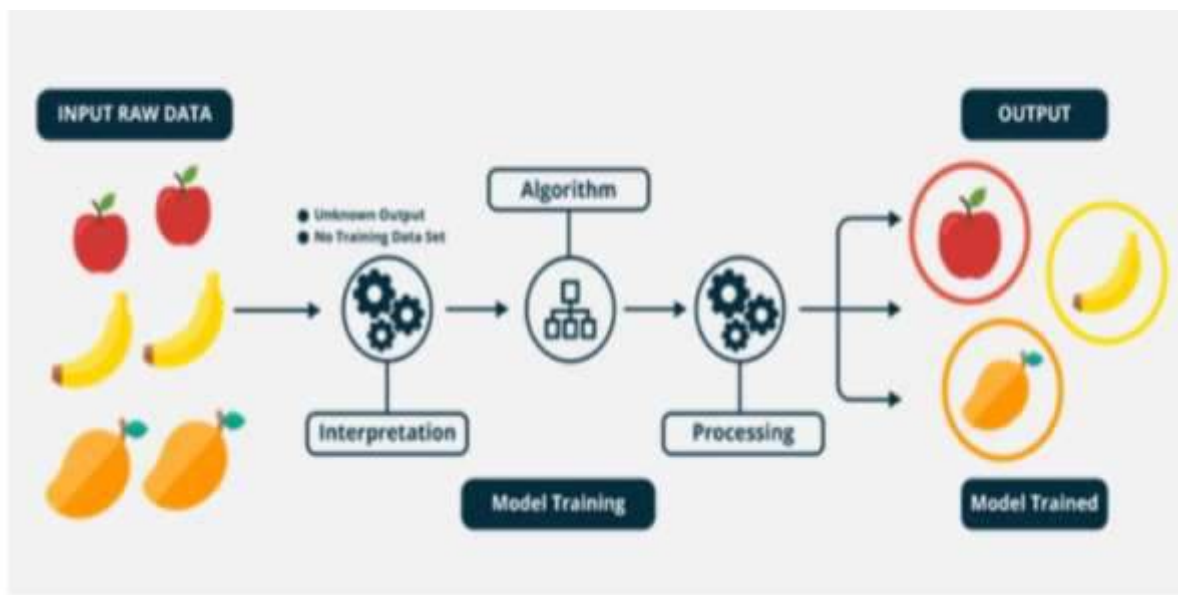


Рис. 9. Модель машинного обучения

В то время как машинное обучение является подмножеством искусственного интеллекта, глубокое обучение является специализированным подмножеством машинного обучения.

Глубокое обучение основывается на алгоритмах машинного обучения, которые основываются на структуре и функциях мозга, и эти алгоритмы называются искусственными нейронными сетями. Эти сети предназначены для непрерывного обучения в процессе работы для повышения качества и точности результатов. Такие системы могут обучаться на неструктурированных данных, таких как фотографии, видео и аудиофайлы.

Алгоритмы глубокого обучения напрямую не отображают входные данные в выходные. Вместо этого они полагаются на несколько слоев обработки. Каждый такой слой передает свой вывод следующему слою, который обрабатывает его и передает его следующему. Именно поэтому такая система из многочисленных слоев называется глубоким обучением.

Такие системы могут обучаться на неструктурированных данных, таких как фотографии, видео и аудиофайлы.

При создании алгоритмов глубокого обучения разработчики и инженеры настраивают количество слоев и тип функций, которые соединяют выходы каждого слоя со входами следующего. Затем они обучают модель, предоставляя множество размеченных примеров.

Например, вы дадите алгоритму глубокого изучения тысячи изображений и метки, которые соответствуют содержанию каждого изображения. Алгоритм будет запускать эти примеры через свою многоуровневую нейронную сеть и будет подгонять веса переменных в каждом слое нейронной сети, чтобы иметь возможность обнаруживать общие шаблоны, которые определяют изображения с похожими метками.

Глубокое обучение устраняет одну из основных проблем, с которой сталкивались алгоритмы обучения предыдущего поколения. В то время как эффективность и производительность алгоритмов машинного обучения предыдущего поколения не улучшилась, по мере роста наборов данных, алгоритмы глубокого обучения продолжают улучшаться по мере поступления большего количества данных. Глубокое обучение оказалось очень эффективным при выполнении различных задач, включая распознавание и транскрипцию голоса, распознавание лиц, медицинскую визуализацию и языковой перевод. Глубокое обучение также является одним из основных компонентов беспилотных автомобилей (рисунок 10).

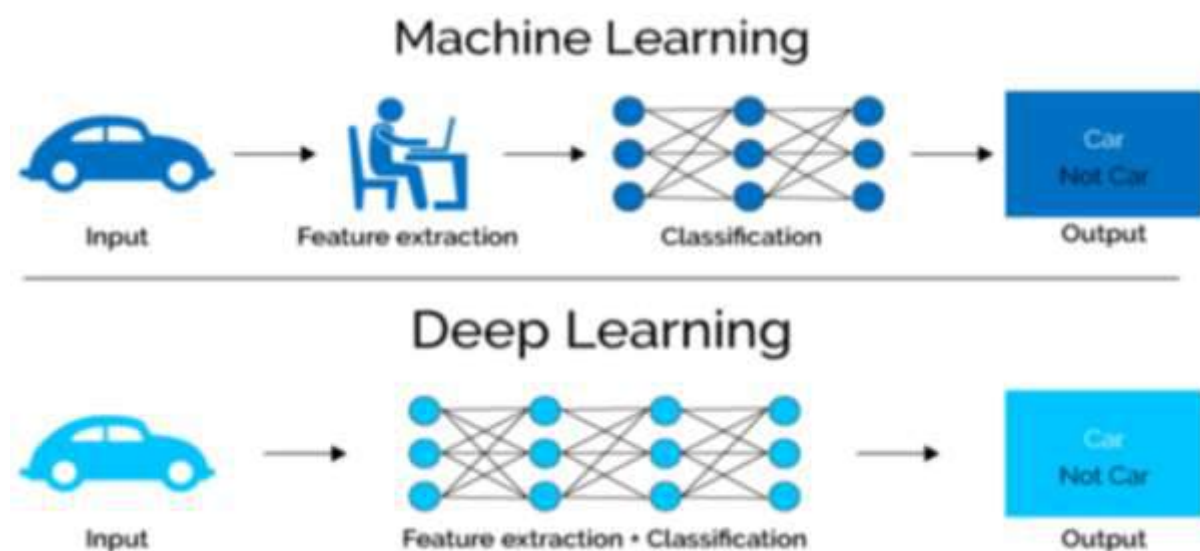


Рис. 10. Структура глубокого обучения.

Искусственная нейронная сеть представляет собой совокупность мелких единиц, называемых нейронами, которые представляют собой

вычислительные единицы, смоделированные по способу обработки информации человеческим мозгом.

Искусственные нейронные сети заимствуют некоторые идеи из биологической нейронной сети мозга, чтобы приблизить некоторые результаты его обработки (рис. 11). Эти единицы или нейроны принимают поступающие данные, также как и биологические нейронные сети, и со временем учатся принимать решения.

Нейронные сети учатся через процесс, называемый обратным распространением. Например, при преобразовании речи в текст, в нейронных сетях вместо кодирования правил вы предоставляете образцы голоса и соответствующий им текст. И нейронная сеть находит общие шаблоны произношения слов, а затем учится сопоставлять новые голосовые записи с соответствующими им текстами. YouTube использует это для автоматического создания субтитров.

Обратное распространение использует набор обучающих данных, которые сопоставляют известные входы с желаемыми выходами. Сначала входы подключаются к сети и определяются выходы. Затем функция ошибки определяет, насколько далеко данный выход находится от желаемого выхода. И наконец, делаются изменения, чтобы уменьшить ошибки.

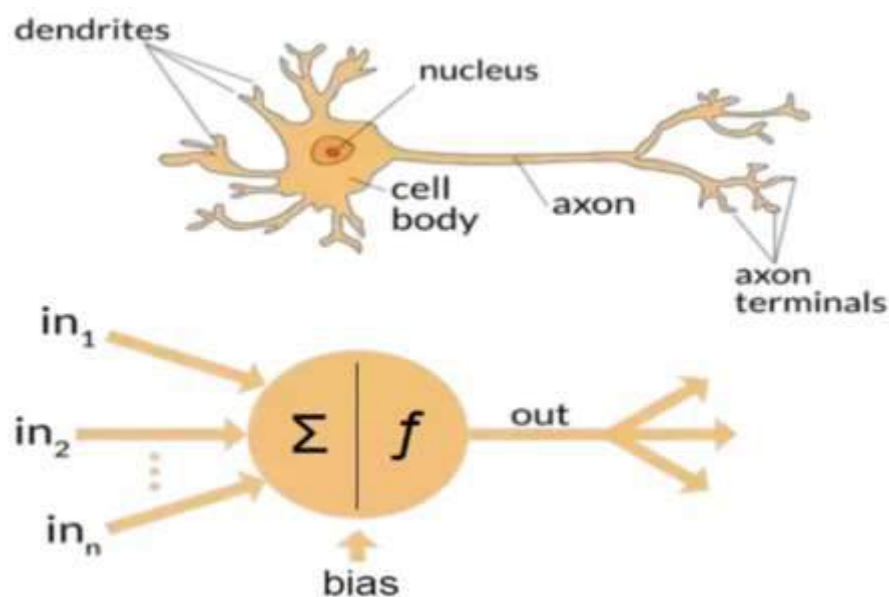


Рис. 11. Модель нейронной сети

Набор нейронов называется слоем, и слой принимает входные данные и обеспечивает выходные данные. Любая нейронная сеть будет

иметь один входной слой и один выходной слой. И нейронная сеть также будет иметь один или несколько скрытых слоев, которые имитируют типы деятельности, происходящих в человеческом мозге.

Скрытые слои принимают набор взвешенных входных данных и выдают результат с помощью функции активации. Нейронная сеть, имеющая более одного скрытого слоя, называется глубокой нейронной сетью (рис. 12).

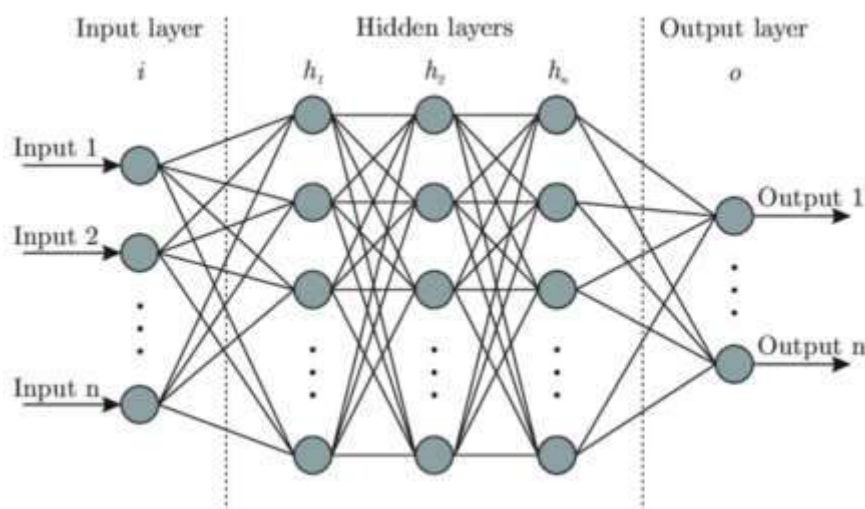


Рис. 12. Глубокая нейронная сеть

Перцептроны – это самые простые и старые типы нейронных сетей. Это однослойные нейронные сети, состоящие из входных узлов, подключенных непосредственно к выходному узлу.

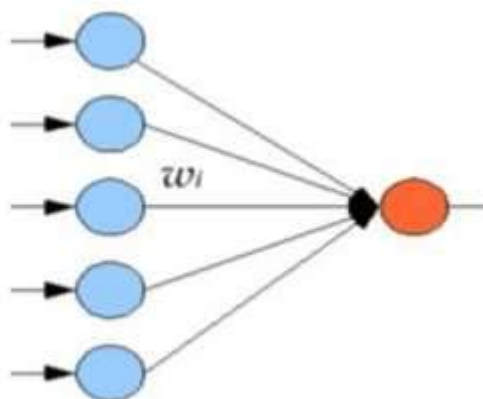


Рис. 13. Перцептрон

Входные слои передают входные значения следующему слою путем умножения на вес и суммирования результатов. Скрытые слои получают

входные данные от других узлов и направляют свои выходные данные на другие узлы (рис. 14).

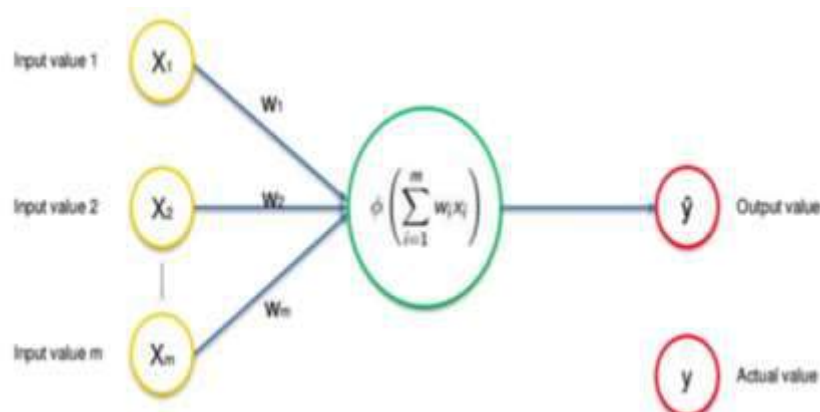


Рис. 14. Схема получения и передачи данных

Скрытые и выходные узлы имеют свойство, называемое смещением bias, которое представляет собой особый тип веса, который применяется к узлу после рассмотрения других входных данных (рис. 15).

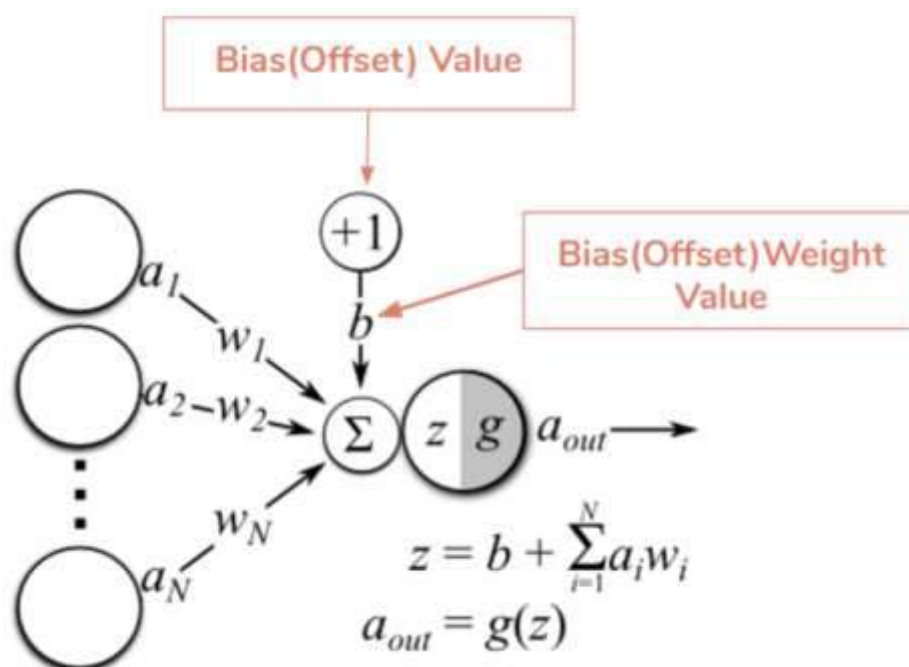


Рис. 15. Свойство смещения

И наконец, функция активации определяет, как узел реагирует на свои входные данные. Функция запускается на сумме входов и смещения, а затем результат передается как выходной. Функции активации могут

принимать различные формы, и их выбор является критическим компонентом успеха нейронной сети.

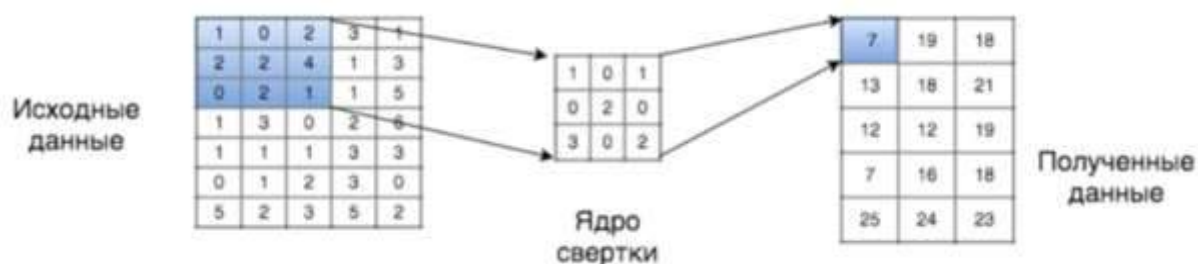


Рис. 16. Сверточные нейронные сети

Сверточные нейронные сети или CNN представляют собой многослойные нейронные сети, которые основываются на работе зрительной коры животных.

CNN полезны в таких приложениях, как обработка изображений, распознавание видео и обработка языка.

Свертка – это математическая операция, в которой функция применяется к другой функции, а результат представляет собой смесь двух функций. Свертки хороши при обнаружении простых структур на изображении и объединении этих простых функций для создания более сложных функций.

В сверточной сети этот процесс происходит в последовательности слоев, каждый из которых проводит свертку на выходе предыдущего слоя. CNN являются экспертами в построении сложных функций из менее сложных.

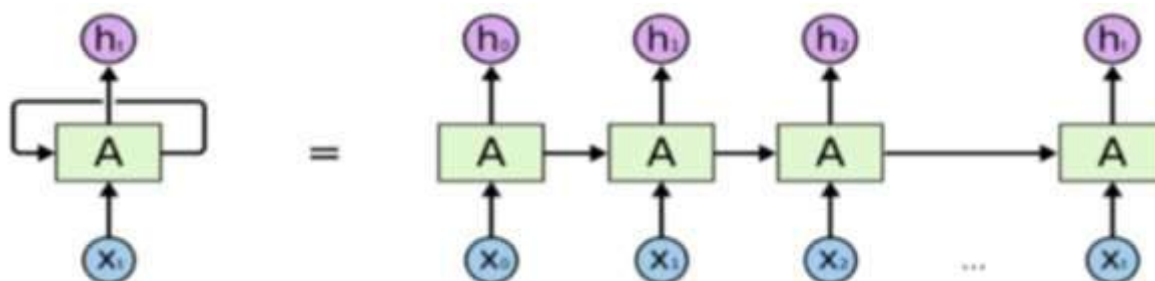


Рис. 17. Рекуррентные сети

Рекуррентные нейронные сети или RNN являются рекуррентными, потому что они выполняют одну и ту же задачу для каждого элемента

последовательности, причем предыдущие выходы питают входы последующих этапов.

В обычной нейронной сети вход обрабатывается через несколько слоев, а выход создается с допущением, что два последовательных входа независимы друг от друга, но это может не выполняться в определенных сценариях.

Например, когда нам нужно учитывать контекст, в котором было произнесено слово, в таких сценариях необходимо учитывать зависимость от предыдущих наблюдений, чтобы получить результат. И RNN могут использовать информацию в длинных последовательностях, причем каждый уровень сети представляет наблюдение в определенное время.

### **Лекция 3. Понятие больших языковых моделей (LLM)**

Большие языковые модели (LLM) – это очень большие модели глубокого обучения, которые предварительно обучены на огромных объемах данных. Лежащий в основе трансформер – это набор нейронных сетей, каждая из которых состоит из кодера и декодера с возможностью самонаблюдения. Кодер и декодер извлекают значения из последовательности текста и понимают отношения между имеющимися в ней словами и фразами.

Трансформеры LLM способны обучаться без наблюдения, хотя точнее будет сказать, что трансформеры осуществляют самообучение. Именно благодаря этому процессу трансформеры учатся понимать базовую грамматику и языки, а также усваивать знания.

В отличие от рекуррентных нейронных сетей (RNN), которые последовательно обрабатывают входные данные, трансформеры обрабатывают целые последовательности параллельно. Это позволяет специалистам по обработке данных использовать графические процессоры для обучения LLM на основе трансформеров, что значительно сокращает время обучения.

Архитектура нейронной сети трансформера позволяет использовать очень большие модели, часто с сотнями миллиардов параметров. Такие сверхбольшие модели могут получать огромные объемы данных, часто из Интернета, а также из таких источников, как индекс Common Crawl,

насчитывающий более 50 миллиардов веб-страниц, и Википедия, насчитывающая около 57 миллионов страниц.

Большие языковые модели невероятно гибкие. Одна модель может выполнять совершенно разные задачи, такие как ответы на вопросы, обобщение документов, языковые переводы и составление предложений.

LLM могут кардинально повлиять на создание контента и использованию людьми поисковых систем и виртуальных помощников.

Хотя LLM и не идеальны, они демонстрируют способность делать точные прогнозы на основе относительно небольшого количества подсказок или входных данных. LLM можно использовать для генеративного искусственного интеллекта для создания контента на основе подсказок к вводу на естественном языке.

LLM большие, очень большие. Они могут учитывать миллиарды параметров и применяться для множества задач. Ниже приведены некоторые примеры.

- Модель GPT-3 Open AI имеет 175 миллиардов параметров. Ее двоюродный брат, ChatGPT, может определять закономерности на основе данных и генерировать естественные и удобочитаемые выходные данные. Хотя нам неизвестен размер Claude 2, он может принимать на входе до 100 тысяч токенов в каждом запросе и, соответственно, работать с сотнями страниц технической документации или даже целой книгой.
- Модель Jurassic-1 от AI21 Labs имеет 178 миллиардов параметров, словарный запас из 250 000 слов и аналогичные разговорные возможности.
- Модель Cohere Command обладает аналогичными возможностями и может работать более чем на 100 разных языках.
- Компания LightOn Paradigm предлагает базовые модели с заявленными возможностями, превосходящими возможности GPT-3. Все эти LLM поставляются с API, которые позволяют разработчикам создавать уникальные приложения для генеративного искусственного интеллекта.

Ключевым фактором в работе LLM является то, как они представляют слова. Более ранние формы машинного обучения использовали числовую таблицу для представления каждого слова. Но эта форма представления не могла распознать взаимосвязи между словами, например имеющими схожие значения. Это ограничение было преодолено

за счет использования многомерных векторов, обычно называемых эмбедингами слов, чтобы слова со схожими контекстными значениями или другими взаимосвязями находились близко друг к другу в векторном пространстве.

Используя эмбединги слов, трансформеры могут предварительно обрабатывать текст в виде числовых представлений через кодировщик и понимать контекст слов и фраз со схожими значениями, а также другие взаимосвязи между словами, например частями речи. Затем LLM могут применять эти знания языка с помощью дешифратора для получения уникальных выходных данных.

Существует множество практических применений для LLM:

1) Копирайтинг. Помимо GPT-3 и ChatGPT, Claude, Llama 2, Cohere Command и Jurassicai могут написать оригинальную копию. AI21 Wordspice предлагает внести изменения в оригинальные предложения для улучшения стиля и голоса.

2) Ответы в базе знаний. Этот метод, часто называемый наукоёмкой обработкой естественного языка (KI-NLP), относится к LLM, которые могут отвечать на конкретные вопросы, содержащиеся в справочной информации в цифровых архивах. Примером может служить способность игровой площадки AI21 Studio отвечать на общие вопросы.

3) Классификация текста. Используя кластеризацию, LLM могут классифицировать текст со схожими значениями или смыслом. Варианты применения включают измерение настроений клиентов, определение взаимосвязи между текстами и поиск документов.

4) Генерация кода. LLM хорошо разбираются в генерации кода на основе запросов на естественном языке. В качестве примеров можно привести Amazon CodeWhisperer и кодексы Open AI, используемый в GitHub Copilot, который может писать код на Python, JavaScript, Ruby и некоторых других языках программирования. Другие приложения для программирования включают создание SQL-запросов, написание команд командной строки и дизайн веб-сайтов.

5) Генерация текста. Как и в случае с генерацией кода, при генерации текста можно завершить неполные предложения, написать документацию по продукту или, как в случае с Alexa Create, написать небольшой детский рассказ.

Нейронные сети на основе трансформеров очень большие. Эти сети содержат несколько узлов и уровней. Каждый узел в уровне связан со

всеми узлами последующего уровня, каждый из которых имеет вес и смещение. Веса и смещения, а также эмбединги называются параметрами модели. Большие нейронные сети на основе трансформеров могут иметь миллиарды и миллиарды параметров. Размер модели обычно определяется эмпирической зависимостью между размером модели, количеством параметров и размером обучающих данных.

Обучение проводится с использованием большого массива высококачественных данных. Во время обучения модель итеративно корректирует значения параметров до тех пор, пока модель правильно не предскажет следующий токен из предыдущей последовательности входных токенов. Это достигается с помощью методов самообучения, которые учат модель настраивать параметры, чтобы максимально повысить вероятность появления следующих токенов в учебных примерах. Обученные LLM можно легко адаптировать к выполнению нескольких задач с использованием относительно небольших наборов контролируемых данных. Этот процесс называется точной настройкой.

Существуют три распространенные модели обучения:

- Обучение без примеров: базовые LLM могут реагировать на широкий спектр запросов без специального обучения, часто с помощью подсказок, хотя точность ответов различается.
- Обучение в несколько приемов: приведя несколько подходящих примеров обучения, можно значительно повысить производительность базовой модели в конкретной области.
- Точная настройка: это расширение обучения в несколько приемов, в ходе которой специалисты по анализу данных обучают базовую модель корректировать ее параметры с помощью дополнительных данных, относящихся к конкретному применению.

#### ● Лекция 4. Генеративный ИИ на основе LLM и его применение

*Генеративный искусственный интеллект (генеративный ИИ)* – это тип ИИ, который может создавать новый контент и идеи, включая разговоры, истории, изображения, видео и музыку. Технологии искусственного интеллекта пытаются имитировать человеческий интеллект в таких нетрадиционных вычислительных задачах, как распознавание изображений, обработка естественного языка (NLP) и перевод. Генеративный искусственный интеллект – это следующий шаг в разработке искусственного интеллекта. Вы можете научить его изучать

человеческий язык, языки программирования, искусство, химию, биологию и любые сложные предметы. Он использует изученные данные для решения новых задач. Например, он может выучить английскую лексику и создавать из нее стихотворения. Ваша организация может использовать генеративный искусственный интеллект для различных целей, таких как чат-боты, создание мультимедийных материалов, разработка и дизайн продуктов.

Приложения на основе генеративного искусственного интеллекта, такие как ChatGPT, привлекают всеобщее внимание и поражают воображение. С их помощью можно переосмыслить большинство процессов взаимодействия с клиентами и разработать новые, невиданные ранее приложения, а также помочь клиентам достичь новых показателей производительности.

По данным компании Goldman Sachs, генеративный ИИ может обеспечить увеличение мирового валового внутреннего продукта (ВВП) на 7 % (или почти на 7 трлн долл.). Также предполагается, что за 10 лет рост продуктивности может увеличиться на 1,5 процентных пункта.

Приведем несколько преимуществ генеративного ИИ:

1) Ускорение исследований. Алгоритмы генеративного ИИ могут по-новому исследовать и анализировать сложные данные. Таким образом, исследователи могут обнаруживать новые тенденции и закономерности, которые в других случаях могут быть не очевидны. Эти алгоритмы способны обобщать информацию, описывать несколько путей решения, генерировать идеи и создавать подробную документацию на основе записей исследований. Именно поэтому генеративный ИИ значительно повышает эффективность исследований и инноваций.

Например, системы генеративного ИИ используются в фармацевтической промышленности для генерации и оптимизации белковых последовательностей, а также значительного ускорения создания лекарственных препаратов.

2) Повышение качества обслуживания клиентов. Генеративный ИИ может естественным образом реагировать на разговор с человеком, а также служить инструментом для обслуживания клиентов и персонализации их рабочих процессов.

Например, можно использовать чат-боты, голосовые боты и виртуальные помощники на базе искусственного интеллекта, которые точнее реагируют на запросы клиентов при первом обращении. Благодаря

персонализированной подаче предложений и общению с клиентами можно добиться большей вовлеченности потребителей.

3) Оптимизация бизнес-процессов. С помощью генеративного ИИ можно оптимизировать бизнес-процессы, используя приложения машинного обучения и ИИ во всех сферах деятельности. Эту технологию можно применять во всех сферах бизнеса, включая проектирование, маркетинг, обслуживание клиентов, финансы и продажи.

К примеру, генеративный ИИ можно использовать при:

- извлечении и обобщении данных из любых источников для использования функций поиска знаний;
- оценке и оптимизации различных сценариев снижения затрат в таких областях, как маркетинг, реклама, финансы и логистика;
- генерации синтетических данных и создании маркированных данных для обучения под наблюдением и других процессов машинного обучения.

4) Увеличение продуктивности работы персонала. Модели генеративного ИИ могут дополнять рабочие процессы сотрудников и выполнять функции эффективных помощников для всех сотрудников вашей организации. От поиска информации до ее создания – они могут делать все подобно человеку.

С помощью генеративного ИИ можно повысить производительность труда различных категорий персонала, поскольку он:

- поддерживает творческие задачи за счет генерации нескольких прототипов на основе определенных входных данных и ограничений; оптимизирует существующие проекты на основе отзывов людей и определенных ограничений;
- создает новые предложения по программному коду для задач разработки приложений;
- помогает руководителям в составлении отчетов, сводок и прогнозов;
- разрабатывает новые скрипты продаж, контент для электронной почты и блогов для сотрудников отдела маркетинга.

Как и любой другой искусственный интеллект, генеративный ИИ работает на основе моделей машинного обучения – очень больших моделей, предварительно обученных на огромном количестве данных.

Базовые модели (FM) – это модели машинного обучения, обученные на широком спектре обобщенных и немаркированных данных. Они способны выполнять множество общих задач.

Базовые модели – это результат последних достижений в технологии, которая развивалась десятилетиями. Как правило, FM использует изученные закономерности и взаимосвязи для прогнозирования следующего элемента последовательности.

Например, при создании изображения модель его анализирует и создает более резкую и четкую версию. Аналогично, в случае с текстом модель предсказывает следующее слово в текстовой строке на основе предыдущих слов и их контекста. Затем она выбирает следующее слово, используя методы вероятностного распределения.

Большие языковые модели (LLM) – это один из классов базовых моделей. Например, генеративные предварительно обученные модели-трансформеры (GPT), разработанные OpenAI являются LLM. LLM специально ориентированы на языковые задачи, такие как обобщение, генерация текста, классификация, открытая беседа и извлечение информации.

Особенностью LLM является их способность выполнять несколько задач. Это возможно благодаря тому, что в них присутствуют многие свойства, позволяющие осваивать сложные концепции.

LLM, такой как GPT-3, может учитывать миллиарды параметров и генерировать контент, используя небольшое количество входных данных. Кроме того, изучая данные в масштабе Интернета во всех их различных формах и множестве закономерностей, большие языковые модели учатся применять свои знания в самых разных контекстах.

Традиционные модели машинного обучения были дискриминативными или концентрировались на классификации точек данных. С их помощью пытались определить взаимосвязь между известными и неизвестными факторами, например, рассматривая изображения (расположение пикселей, линии, цвет и форму), то есть известные данные, и сопоставляя их с неизвестными – словами. Математически модели работали путем определения уравнений, в которых можно было численно отобразить неизвестные и известные факторы в виде переменных  $x$  и  $y$ .

Генеративные модели упрощают эти процессы. Вместо того чтобы предсказывать метку по некоторым признакам, они пытаются предсказать признаки по определенной метке. Математически генеративное моделирование рассчитывает вероятность того, что  $x$  и  $y$  совпадут. Оно

изучает распределение различных характеристик данных и их взаимосвязь.

Например, генеративные модели анализируют изображения животных и регистрируют такие переменные, как различные формы ушей, глаз, хвостов и кожи. Они изучают признаки и их взаимосвязи, чтобы понять, как выглядят различные животные в целом. Затем они могут создавать новые изображения животных, которых не было в наборе обучающих данных.

Приведем примеры нескольких широких категорий моделей генеративного искусственного интеллекта.

Модели диффузии (рис.18). Модели диффузии создают новые данные путем итеративного внесения контролируемых случайных изменений в исходную выборку данных. Они начинают с исходных данных и постепенно уменьшают их сходство с оригиналом, добавляя незначительные изменения (шум). Этот шум тщательно контролируется, чтобы генерируемые данные сохраняли целостность и реалистичность.

После добавления шума в течение нескольких итераций модель диффузии изменяет процесс на обратный, постепенно удаляя шум, в результате чего получается новый образец данных, похожий на исходный.

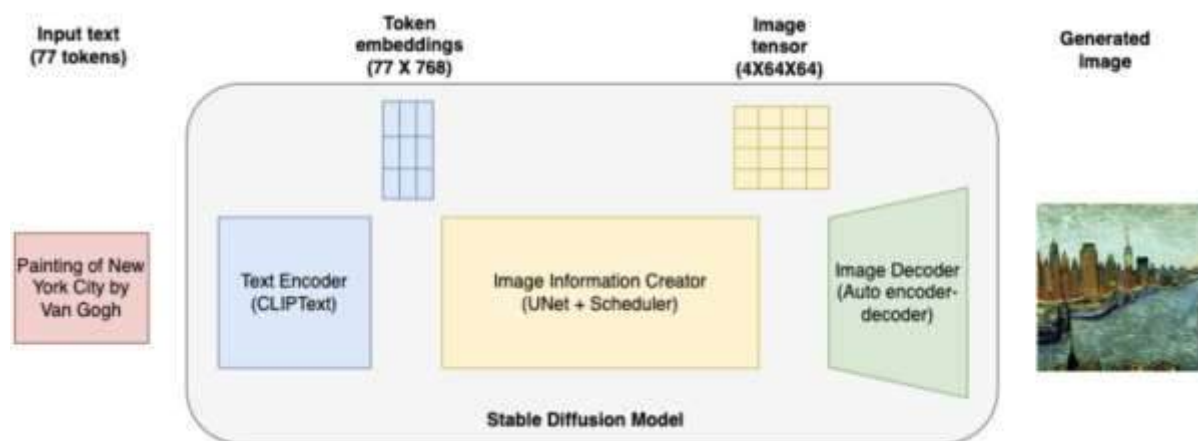


Рис. 18. Модели диффузии

Генеративные состязательные сети. Генеративная состязательная сеть (GAN) – еще одна модель генеративного ИИ, развивающая концепцию модели диффузии.

Модели GAN обучают две нейронные сети в соревновательном режиме. Первая сеть, известная как генератор, создает поддельные образцы данных, добавляя в них случайный шум. Вторая сеть, называемая

дискриминатором, пытается отличить реальные данные от поддельных, создаваемых генератором.

В процессе обучения генератор постоянно совершенствует способность создавать реалистичные данные, а дискриминатор все лучше отличает реальные данные от поддельных. Этот состязательный процесс продолжается до тех пор, пока генератор не выдаст настолько убедительные данные, что дискриминатор не сможет отличить их от реальных.

Модели GAN широко используются в задачах генерации реалистичных изображений, передачи стиля и дополнения данных.

Вариационные автокодировщики (рис. 19). Вариационные автокодировщики (VAE) обучаются компактному представлению данных, называемому скрытым пространством. Скрытое пространство – это математическое представление данных. Его можно рассматривать как уникальный код, представляющий данные на основе всех их атрибутов. Например, при изучении лиц в скрытом пространстве присутствуют числа, представляющие форму глаз, носа, скул и ушей.

VAE используют две нейронные сети – кодировщик и дешифратор. Нейронная сеть-кодировщик сопоставляет входные данные со средним значением и дисперсией для каждого измерения скрытого пространства. Она генерирует случайную выборку из гауссовского (нормального) распределения. Эта выборка является точкой в скрытом пространстве и представляет собой сжатую упрощенную версию входных данных.

Нейронная сеть-дешифратор получает эту выборку из скрытого пространства и преобразует ее в данные, напоминающие исходный входной сигнал. Для измерения степени соответствия реконструированных данных исходным используются математические функции.

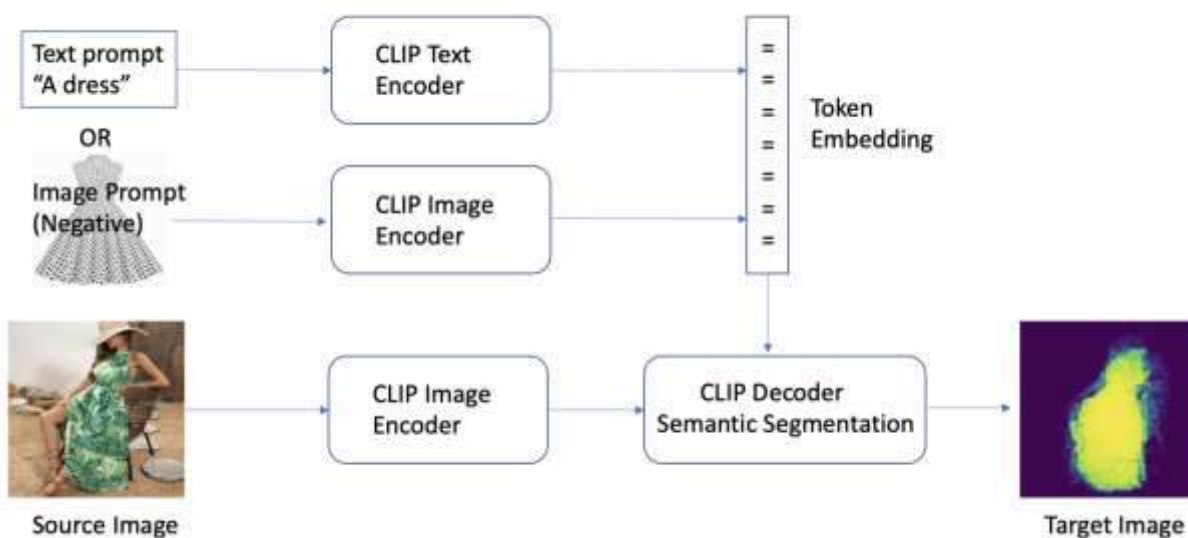


Рис. 19. Вариационные автокодировщики

Модели на основе трансформеров. Модель генеративного ИИ на основе трансформера опирается на концепцию кодировщика и дешифратора VAE. Модели, основанные на трансформерах, добавляют новые уровни к кодировщику, чтобы повысить производительность при выполнении текстовых задач, в том числе при понимании, переводе и написании творческих работ.

Модели на основе трансформеров используют механизм самовнимания. Они оценивают важность различных частей последовательности входных данных при обработке каждого элемента этой последовательности.

Еще одна ключевая особенность этих моделей искусственного интеллекта заключается в реализации контекстного встраивания. Кодировка элемента последовательности зависит не только от самого элемента, но и от его контекста в последовательности.

Как работают модели на основе трансформеров? Чтобы понять, как работают модели на основе трансформеров, представьте предложение как последовательность слов.

Самовнимание помогает модели сосредоточиться на важности слов при обработке каждого из них. Для того чтобы передать различные типы связей между словами, в генеративной модели на основе трансформера используется несколько слоев кодировщика, называемых головками внимания. Каждая головка учится воспринимать различные части

последовательности входных сигналов, чтобы модель могла одновременно учитывать различные аспекты данных.

Каждый слой также уточняет контекстные встраивания. Благодаря слоям контекстные встраивания становятся более информативными и передают различные значения – от грамматического синтаксиса до сложных семантических смыслов.

Как развивалась технология генеративного искусственного интеллекта? Примитивные генеративные модели уже несколько десятилетий используются в статистике для анализа числовых данных. Нейронные сети и глубокое обучение были предшественниками современного генеративного ИИ. Вариационные автокодировщики, разработанные в 2013 году, стали первыми глубокими генеративными моделями, способными генерировать реалистичные изображения и речь.

В вариационных автокодировщиках была предусмотрена возможность создания новых вариаций нескольких типов данных, в результате чего быстро появились другие генеративные модели ИИ, например генеративные состязательные сети и диффузионные модели. С помощью этих инноваций генерировали данные, которые, будучи созданными искусственно, все больше напоминали реальные.

В 2017 году в исследованиях ИИ произошел очередной поворот – появились трансформеры. Они органично объединили архитектуру кодировщика и дешифратора с механизмом внимания и оптимизировали процесс обучения языковых моделей, обеспечив исключительную эффективность и универсальность. Такие известные модели, как GPT, возникли как базовые модели, способные к предварительному обучению на обширных массивах необработанных текстов и точной настройке для решения различных задач.

Трансформеры расширили возможности обработки естественного языка и генерации для решения различных задач, от перевода и обобщения до ответов на вопросы.

Многие генеративные модели искусственного интеллекта продолжают демонстрировать значительные успехи и используются во многих отраслях. Недавние инновации направлены на совершенствование моделей для работы с собственными данными. Исследователи также хотят создавать тексты, изображения, видео и речь, все более и более похожие на человеческие.

Как генеративный ИИ повлияет на отрасли? Хотя генеративный ИИ со временем сможет повлиять на все сферы деятельности, некоторые из них могут быстро извлечь выгоду из этой технологии.

Финансовые сервисы. Компании, предоставляющие финансовые услуги, могут использовать возможности генеративного ИИ, чтобы лучше обслуживать своих клиентов и при этом сокращать расходы.

- Финансовые компании могут использовать чат-боты для создания рекомендаций по продуктам и ответов на запросы клиентов, чтобы повысить уровень обслуживания.
- Кредитные учреждения могут ускорить выдачи займов на рынках с недостаточным финансовым обслуживанием, особенно в развивающихся странах.
- Банки могут быстро обнаруживать мошенничество в претензиях, кредитных картах или займах.
- Инвестиционные компании могут использовать возможности генеративного ИИ для предоставления своим клиентам безопасных и персонализированных финансовых консультаций по низким ценам.

Здравоохранение и медико-биологические разработки. Один из наиболее перспективных вариантов использования генеративного искусственного интеллекта – ускорение разработки и исследования лекарств. Генеративный ИИ использует модели для создания новых белковых последовательностей со специфическими свойствами, предназначенных для разработки антител, ферментов, вакцин, а также генной терапии.

Компании, занимающиеся здравоохранением и медико-биологическими науками, могут использовать генеративные модели, чтобы разрабатывать синтетические последовательности генов для применения в синтетической биологии и метаболической инженерии. Например, они могут создавать новые биосинтетические пути или оптимизировать экспрессию генов для целей биопроизводства.

Наконец, генеративный искусственный интеллект можно использовать для создания синтетических данных о пациентах и медицинских учреждениях. Это полезно при обучении моделей искусственного интеллекта, моделировании клинических испытаний или изучении редких заболеваний, когда доступ к большим массивам данных реального мира отсутствует.

Автомобильная промышленность и производство. Автомобильные компании могут использовать технологии генеративного ИИ для различных целей, начиная от проектирования автомобилей и заканчивая обслуживанием клиентов. Например, они могут оптимизировать конструкцию механических частей таким образом, чтобы уменьшить аэродинамическое сопротивление в конструкции автомобиля, или адаптировать дизайн персональных помощников.

Автомобильные компании используют генеративный искусственный интеллект для улучшения обслуживания клиентов, быстро отвечая на их наиболее распространенные вопросы. С помощью генеративного искусственного интеллекта можно оптимизировать производственные процессы и снизить затраты, создавая новые конструкции материалов, микросхем и деталей.

Генеративный искусственный интеллект также можно использовать для того, чтобы генерировать синтетические данные для тестирования приложений. Это особенно полезно для данных, которые редко включаются в наборы тестовых данных (например, дефекты или крайние случаи).

Мультимедиа и развлечения. Модели генеративного ИИ могут создавать новый контент, от анимации и сценариев до полнометражных фильмов, с меньшими затратами средств и времени, чем при традиционном производстве.

Приведем другие способы использования генеративного ИИ в отрасли:

- Артисты могут дополнять и улучшать свои альбомы музыкой, сгенерированной искусственным интеллектом, для создания совершенно новых жанров.
- Медийные организации могут использовать искусственный интеллект, чтобы впечатлить аудиторию, предлагая персонализированный контент и рекламу для увеличения доходов.
- Игровые компании могут использовать генеративный искусственный интеллект для создания новых игр и предоставления игрокам возможности создавать аватары.

Телекоммуникации. Первые примеры использования генеративного ИИ в телекоммуникациях касаются пересмотра качества взаимодействия с клиентами. Опыт работы с клиентами определяется совокупным

взаимодействием подписчиков во всех точках взаимодействия с клиентами.

Например, компании, работающие в сфере телекоммуникаций, могут применять генеративный ИИ для улучшения качества обслуживания клиентов с помощью разговорных агентов, похожих на живых людей. Они также могут оптимизировать производительность сети, анализируя сетевые данные и разрабатывая рекомендации по устранению неполадок. Кроме того, они могут по-новому выстраивать отношения с клиентами с помощью персональных продавцов-консультантов, которые работают с людьми один на один.

**Энергетика.** Генеративный искусственный интеллект подходит для задач энергетического сектора, связанных со сложным анализом необработанных данных, распознаванием образов, прогнозированием и оптимизацией. Компании, предоставляющие услуги в сфере энергетики, могут повысить качество обслуживания клиентов, анализируя данные предприятия для выявления закономерностей использования. На основе этой информации они могут разрабатывать целевые предложения продуктов, программы энергоэффективности или инициативы по реагированию на спрос.

Генеративный искусственный интеллект может помочь в управлении сетями, повышении безопасности эксплуатации объектов и оптимизации производства энергии с помощью моделирования резервуаров.

Что такое GPT? Генеративные предобученные трансформеры, широко известные как GPT, представляют собой семейство моделей нейронных сетей, использующих архитектуру трансформеров и являющихся ключевым достижением в области искусственного интеллекта, с помощью которого работают генеративные приложения ИИ, такие как ChatGPT. Модели GPT дают приложениям возможность генерировать текст и контент (изображения, музыку и многое другое), похожий на созданный человеком, и отвечать на вопросы в разговорной манере. Организации из разных отраслей используют модели GPT и генеративный искусственный интеллект для ботов вопросов и ответов, краткого изложения текста, генерации контента и поиска.

Почему GPT так важен? Модели GPT, в частности используемая в них архитектура трансформера, представляют собой значительный прорыв в исследованиях искусственного интеллекта. Появление моделей GPT

стало переломным моментом в широком распространении машинного обучения, поскольку теперь эту технологию можно использовать для автоматизации и улучшения широкого спектра задач, начиная с перевода на язык и суммирования документов и заканчивая написанием постов в блогах, созданием веб-сайтов, визуальных эффектов, анимации, написанием кода, исследованием сложных тем и даже сочинением стихов. Ценность этих моделей заключается в их скорости и масштабе, в котором они могут работать.

Например, если на исследование, написание и редактирование статьи по ядерной физике может потребоваться несколько часов, модель GPT может создать ее за считанные секунды. Модели GPT послужили толчком к исследованиям в области искусственного интеллекта, направленным на создание общего искусственного интеллекта, а это значит, что машины могут помочь организациям выйти на новый уровень производительности и переосмыслить свои приложения, а также опыт работы с клиентами.

Для чего используют модели GPT? Модели GPT – это языковые модели общего назначения, которые могут выполнять широкий спектр задач: от создания оригинального контента до написания кода, суммирования текста и извлечения данных из документов.

Приведем примеры некоторых способов использования моделей GPT.

1) Создание контента для социальных сетей. Цифровые маркетологи с помощью искусственного интеллекта могут создавать контент для своих кампаний в социальных сетях. Например, они могут предложить модели GPT создать сценарий пояснительного видеоролика. Программное обеспечение для обработки изображений на базе GPT может создавать мемы, видео, маркетинговые тексты и другой контент из текстовых инструкций.

2) Конвертирование текста в разные стили. Модели GPT генерируют текст в повседневном, юмористическом, профессиональном и других стилях. С помощью моделей профессионалы могут переписать определенный текст в другой форме. Например, юристы могут использовать модель GPT для превращения юридических копий в простые пояснительные примечания.

3) Написание и изучение кода. Являясь языковыми моделями, модели GPT могут понимать и писать компьютерный код на различных

языках программирования. Модели могут помогать учащимся, объясняя им компьютерные программы доступным языком. Кроме того, опытные разработчики могут использовать инструменты GPT для автоматического предложения соответствующих фрагментов кода.

4) Анализ данных. Модель GPT может помочь бизнес-аналитикам эффективно собирать большие объемы данных. Языковые модели ищут необходимые данные, рассчитывают и отображают результаты в таблице данных или электронной таблице. Некоторые приложения могут отображать результаты на диаграмме или создавать подробные отчеты.

5) Создание учебных материалов. Преподаватели могут использовать программное обеспечение на основе GPT для создания учебных материалов, таких как тесты и учебные пособия. Точно так же они могут использовать модели GPT для оценки ответов.

6) Создание интерактивных голосовых помощников. Благодаря моделям GPT можно создавать умных интерактивных голосовых помощников. В то время как многие чат-боты реагируют только на базовые устные подсказки, благодаря моделям GPT можно создавать чат-боты с возможностями разговорного искусственного интеллекта. Кроме того, эти чат-боты могут общаться в устной форме, как и люди, в сочетании с другими технологиями искусственного интеллекта.

Как работает GPT? Хотя модели GPT правильно называть искусственным интеллектом (ИИ), это очень общее определение. В частности, модели GPT представляют собой модели языкового прогнозирования на основе нейронных сетей, построенные на архитектуре трансформера. Они анализируют запросы на естественном языке, называемые подсказками, и предсказывают наилучший возможный ответ, основываясь на своем понимании языка.

Для этого модели GPT полагаются на знания, полученные после обучения работе с сотнями миллиардов параметров в огромных наборах языковых данных. Они могут учитывать контекст ввода и динамически обрабатывать различные части входных данных, чтобы генерировать длинные ответы, а не просто следующее слово в последовательности. Например, если попросить создать фрагмент контента, основанного на творчестве Шекспира, модель GPT делает это, запоминая и восстанавливая новые фразы и целые предложения с похожим литературным стилем.

Существуют разные типы нейронных сетей, например рекуррентные и сверточные. Модели GPT представляют собой нейронные сети

трансформера. Архитектура нейронной сети трансформера использует механизмы самонаблюдения для концентрации внимания на различных частях входного текста во время каждого этапа обработки. Модель трансформера улавливает больше контекста и повышает производительность при решении задач обработки естественного языка (NLP). Она состоит из двух основных модулей: кодировщик и дешифратор.

**Кодировщик.** Трансформеры предварительно обрабатывают входные текстовые данные в виде вложений, представляющих собой математические представления слова. При кодировании в векторном пространстве ожидается, что слова, расположенные ближе друг к другу, будут ближе по смыслу. Эти вложения обрабатываются компонентом кодировщика, который собирает контекстную информацию из последовательности ввода. Получив входные данные, блок кодирования сети трансформера разделяет слова на вкрапления и присваивает каждому из них вес. Веса – это параметры, указывающие на релевантность слов в предложении.

Кроме того, позиционные кодировщики дают возможность моделям GPT избежать двусмысленных значений, когда слово используется в других частях предложения. Например, позиционное кодирование позволяет модели трансформера различать семантические различия между следующими предложениями.

- Собака гонится за кошкой
- Кошка гонится за собакой

Таким образом, кодировщик обрабатывает входное предложение и генерирует векторное представление фиксированной длины, известное как встраивание. Это представление используется модулем дешифратора.

**Дешифратор** использует векторное представление для прогнозирования требуемого выхода данных. В нем встроены механизмы самонаблюдения, позволяющие сосредоточиться на различных частях входного сигнала и предположить подходящий выходной сигнал. С помощью сложных математических методов дешифратор может оценить несколько различных выходных сигналов и предсказать наиболее точный из них.

По сравнению со своими предшественниками, такими как рекуррентные нейронные сети, трансформеры более пригодны для параллелизации, поскольку они не обрабатывают слова последовательно

по одному за раз, а одновременно обрабатывают весь входной сигнал во время цикла обучения. Благодаря этому, а также тысячам часов, потраченных инженерами на точную настройку и обучение моделей GPT, они способны давать беглые ответы практически на любые ваши запросы.

Как обучали модель GPT-3? В опубликованной научной статье исследователи описали генеративное предварительное обучение как способность обучать языковые модели с немаркированными данными и получать точные прогнозы. Первая модель GPT, GPT-1, была разработана в 2018 году. Модель GPT-4 была представлена в марте 2023 года в качестве преемника GPT-3.

Модель GPT-3 была обучена с использованием более 175 миллиардов параметров или весов. Инженеры обучили ее на более чем 45 терабайтах данных из таких источников, как веб-тексты, Common Crawl, книги и «Википедия». До начала обучения среднее качество наборов данных улучшилось по мере перехода модели с версии 1 на версию 3.

Модель GPT-3 тренировали в режиме обучения с частичным наблюдением. Сначала инженеры машинного обучения снабдили модель глубокого обучения немаркированными обучающими данными. Модели GPT-3 следовало понимать предложения, разбивать их на части и перестраивать в новые предложения. Во время обучения без наблюдения GPT-3 самостоятельно пыталась добиться точных и реалистичных результатов. Затем инженеры машинного обучения уточняли результаты с помощью обучения под наблюдением. Этот процесс известен как обучение с подкреплением с обратной связью от человека (RLHF).

Модели GPT можно использовать без дополнительного обучения или настроить их на нескольких примерах для конкретной задачи.

Какие есть приложения, использующие GPT? С момента своего появления модели GPT способствовали внедрению искусственного интеллекта в многочисленные приложения в различных отраслях промышленности. Ниже приведены некоторые примеры:

- Модели GPT можно использовать для анализа отзывов клиентов и их суммирования в понятный текст. Сначала вы можете собрать данные о настроении клиентов из таких источников, как опросы, обзоры и онлайн-чаты, а затем попросить модель GPT обобщить эти данные.

- С помощью моделей GPT можно обеспечить естественное общение виртуальных персонажей с игроками-людьми в виртуальной реальности.
- Модели GPT можно использовать, чтобы упростить поиск для сотрудников справочной службы. Они могут запрашивать базу знаний о продукте на разговорном языке, чтобы получить соответствующую информацию о продукте.

## **Лекция 5. Промпт-инжиниринг: понятие и основные принципы**

Промпт-инжиниринг - это относительно новая дисциплина разработки и оптимизации промптов для эффективного использования языковых моделей (LM) в широком спектре приложений и исследовательских тем. Навыки промт-инжиниринга помогают лучше понять возможности и ограничения больших языковых моделей (LLM).

Исследователи используют промт-инжиниринг для улучшения возможностей LLM на широком спектре общих и сложных задач, таких как вопросно-ответная система и арифметическое рассуждение. Разработчики используют промт-инжиниринг для разработки надежных и эффективных методов промптинга, взаимодействующих с LLM и другими инструментами.

Промт-инжиниринг - это не только разработка и создание промптов. Это охватывает широкий спектр навыков и техник, полезных для взаимодействия и разработки с LLM. Это важный навык для взаимодействия, создания и понимания возможностей LLM. Вы можете использовать промт-инжиниринг для повышения безопасности LLM и создания новых возможностей, таких как добавление доменных знаний и внешних инструментов к LLM.

При работе с промптами вы взаимодействуете с LLM с помощью API или непосредственно. Вы можете настроить несколько параметров, чтобы получить различные результаты для ваших промптов.

### **Настройки LLM**

Температура — Вкратце, чем ниже значение температуры, тем более детерминированными будут результаты в смысле того, что будет выбран самый вероятный следующий токен. Увеличение температуры может привести к большей случайности, что способствует более разнообразным или творческим результатам. Вы фактически увеличиваете веса других

возможных токенов. В плане применения, для задач, связанных с ответами на вопросы на основе фактов, рекомендуется использовать более низкое значение температуры, чтобы стимулировать более точные и краткие ответы. Для генерации стихов или других творческих задач может быть полезно увеличить значение температуры.

**Тор\_р** — Аналогично, с помощью **tor\_p**, техники сэмплирования с использованием температуры, называемой сэмплированием ядра, вы можете контролировать, насколько детерминированной будет модель в генерации ответа. Если вы ищете точные и фактические ответы, установите низкое значение. Если вы ищете более разнообразные ответы, увеличьте значение. Общая рекомендация заключается в том, чтобы изменять только один параметр, а не оба.

**Максимальная длина** — Вы можете управлять количеством токенов, генерируемых моделью, регулируя максимальную длину. Указание максимальной длины помогает предотвратить длинные или ненужные ответы и контролировать затраты.

**Стоп-последовательности** — это строка, которая останавливает модель от генерации токенов. Указание последовательности остановки — это еще один способ контролировать длину и структуру ответа модели. Например, вы можете указать модели генерировать списки, содержащие не более 10 элементов, добавив «11» в качестве стоп-последовательности.

**Штраф за частоту** — Штраф за частоту накладывает штраф на следующий токен, пропорциональный тому, сколько раз этот токен уже появлялся в ответе и подсказке. Чем выше штраф за частоту, тем меньше вероятность того, что слово появится снова. Этот параметр уменьшает повторение слов в ответе модели, назначая более высокий штраф за лексемы, которые кажутся более выраженными. Эта настройка уменьшает повторение слов в ответе модели, назначая более высокий штраф за токены, которые появляются чаще.

**Штраф за наличие** — Штраф за наличие также применяет штраф к повторяющимся токенам, но, в отличие от штрафа за частоту, штраф одинаков для всех повторяющихся токенов. Токен, который появляется дважды, и токен, который появляется 10 раз, наказываются одинаково. Эта настройка останавливает модель от слишком частого повторения фразы в ответе. Если вы хотите, чтобы модель генерировала разнообразный или креативный текст, вы можете использовать более высокий штраф за

наличие. Или, если вам нужно, чтобы модель оставалась сосредоточенной, попробуйте использовать более низкий штраф за наличие.

Как и в случае с температурой и `top_p`, рекомендуется — изменить штраф за частоту или присутствие, но не то и другое одновременно.

## **Основы промптинга**

### **Основные промпты**

Вы можете достичь многого с помощью простых промптов, но качество результатов зависит от того, сколько информации вы предоставляете и насколько хорошо он составлен. Промпт может содержать информацию, такую как инструкция или вопрос, который вы передаете модели, и включать другие детали, такие как контекст, входные данные или примеры. Вы можете использовать эти элементы, чтобы лучше указать модели, что от нее требуется, и в результате получить лучшие результаты.

Давайте начнем с примера простого промпта:

*Промпт*

*The sky is*

*Результат:*

*blue*

*The sky is blue on a clear day. On a cloudy day, the sky may be gray or white.*

Как видите, языковая модель выводит продолжение строк, которое имеет смысл в контексте "The sky is". Результат может быть неожиданным или далеким от задачи, которую вы хотите выполнить.

Этот простой пример также подчеркивает необходимость предоставления большего контекста или инструкций о том, что именно вы хотите достичь.

Давайте попробуем немного улучшить его:

*Промпт:*

*Complete the sentence:*

*The sky is*

*Результат:*

*so beautiful today.*

Уже лучше, не так ли? Вы сказали модели завершить предложение, поэтому результат выглядит гораздо лучше, так как он точно следует вашей инструкции ("Complete the sentence"). Такой подход к

проектированию оптимальных промптов для указания модели выполнения задачи называется промпт инжинирингом.

Приведенный выше пример - это базовая иллюстрация того, что сегодня возможно с помощью LLM (Large Language Models). Современные LLM могут выполнять самые разные задачи, начиная от суммирования текста и математического рассуждения, и заканчивая генерацией кода.

#### Форматирование промптов

Вы попробовали очень простой промпт выше. Стандартный формат промпта имеет следующую структуру:

`<Question>?`

*или*

`<Instruction>`

Вы можете отформатировать его в формат вопроса и ответа (QA), который является стандартным для многих наборов данных QA, следующим образом:

Q: `<Question>?`

A:

При применении такого форматирования промпта это также называется промптингом без примеров (zero-shot prompting), то есть вы напрямую запрашиваете ответ у модели без предоставления каких-либо примеров или демонстраций задачи, которую вы хотите выполнить. Некоторые большие языковые модели имеют возможность выполнять промптинг без примеров, но это зависит от сложности и знания задачи.

Учитывая стандартный формат выше, одна популярная и эффективная техника промптинга называется промптинг с несколькими примерами (few-shot prompting), где вы предоставляете примеры (т.е. демонстрации). Вы можете отформатировать промпты с несколькими примерами следующим образом:

`<Question>?`

`<Answer>`

`<Question>?`

`<Answer>`

`<Question>?`

`<Answer>`

`<Question>?`

В версии в формате вопроса и ответа (QA) это будет выглядеть так:

Q: <Question>?

A: <Answer>

Q: <Question>?

A: <Answer>

Q: <Question>?

A: <Answer>

Q: <Question>?

A:

Имейте в виду, что использование формата вопроса и ответа (QA) не является обязательным. Формат промпта зависит от задачи. Например, вы можете выполнить простую задачу классификации и дать образцы, демонстрирующие задачу, следующим образом:

*Промпт:*

*This is awesome! // Positive*

*This is bad! // Negative*

*Wow that movie was rad! // Positive*

*What a horrible show! //*

*Результат:*

*Negative*

Промпты с несколькими примерами позволяют учиться в контексте, что означает, что языковые модели могут обучаться задачам на основе нескольких демонстраций.

### **Элементы промпта**

По мере того как мы рассматриваем все больше примеров и применений инженерии промптов, вы заметите, что определенные элементы составляют промпт.

Промпт содержит любой из следующих элементов:

Инструкция - конкретная задача или инструкция, которую вы хотите, чтобы модель выполнила

Контекст - внешняя информация или дополнительный контекст, который может направить модель на более точные ответы

Входные данные - вход или вопрос, на который мы хотим найти ответ

Индикатор вывода - тип или формат вывода.

Вам не нужны все четыре элемента для промпта, и формат зависит от задачи. Мы рассмотрим более конкретные примеры в грядущих руководствах.

## **Общие рекомендации по разработке промптов**

Вот несколько советов, которые стоит иметь в виду при разработке ваших промптов:

### ***Начните с простого***

При разработке промптов стоит помнить, что это итеративный процесс, требующий много экспериментов для достижения оптимальных результатов. Хорошей отправной точкой является использование простого плейграунда от OpenAI или Cohere.

Вы можете начать с простых промптов и добавлять больше элементов и контекста по мере стремления к лучшим результатам. Важно итеративно улучшать ваш промпт по мере продвижения. При чтении руководства вы увидите множество примеров, где специфичность, простота и краткость часто дадут вам лучшие результаты.

Когда у вас есть большая задача, включающая множество различных подзадач, вы можете попробовать разбить задачу на более простые подзадачи и постепенно улучшать результаты. Это позволяет избежать слишком большой сложности в процессе проектирования промпта с самого начала.

### ***Инструкция***

Вы можете разрабатывать эффективные промпты для различных простых задач, используя команды для указания модели, что вы хотите достичь, такие как "Write", "Classify", "Summarize", "Translate", "Order", и т.д.

Имейте в виду, что вам также нужно много экспериментировать, чтобы увидеть, что работает лучше всего. Попробуйте разные инструкции с разными ключевыми словами, контекстами и данными, и посмотрите, что работает лучше всего для вашего конкретного случая использования и задачи. Обычно чем более специфичным и соответствующим контекст для задачи, которую вы пытаетесь выполнить, тем лучше. Мы затронем важность выборки и добавления большего контекста в следующих руководствах.

Другие рекомендуют размещать инструкции в начале промпта. Еще одна рекомендация состоит в использовании некоторого ясного разделителя, например "###", для отделения инструкции и контекста.

Например:

*Промпт:*

**### Instruction ###**

*Translate the text below to Spanish:*

*Text: "hello!"*

*Результат:*

*¡Hola!*

### ***Специфичность***

Будьте очень конкретными при написании инструкции и задачи, которую вы хотите, чтобы модель выполнила. Чем более подробным и детальным будет промпт, тем лучше будут и результаты. Это особенно важно, когда вы уже понимаете какого результата или стиля генерации вы хотите добиться. Нет конкретных токенов или ключевых слов, которые приводят к хорошим результатам. Гораздо важнее хороший формат и описательный промпт. Использование примеров в промпте очень эффективно для получения желаемого вывода в конкретных форматах.

При разработке промптов стоит также учитывать длину промпта, так как у неё есть ограничения. Подумайте о том, насколько конкретным и детальным вы хотите быть. Включение слишком многих дополнительных деталей не всегда является хорошим подходом. Детали должны быть соответствовать задаче и способствовать её выполнению. Это то, с чем вам придется много экспериментировать. Мы настоятельно рекомендуем проводить много экспериментов и итераций для оптимизации промптов для ваших приложений.

В качестве примера рассмотрим простой промпт для извлечения определенной информации из текста.

*Промпт:*

*Extract the name of places in the following text.*

*Desired format:*

*Place: <comma\_separated\_list\_of\_places>*

*Input: "Although these developments are encouraging to researchers, much is still a mystery. "We often have a black box between the brain and the effect we see in the periphery," says Henrique Veiga-Fernandes, a neuroimmunologist at the Champalimaud Centre for the Unknown in Lisbon. "If we want to use it in the therapeutic context, we actually need to understand the mechanism.""*

*Результат:*

*Place: Champalimaud Centre for the Unknown, Lisbon*

***Избегайте неточностей***

Учитывая вышеуказанные рекомендации о детализации и улучшении формата, легко попасть в ловушку и слишком увлечься идеей быть самым умным при написании промптов, что может привести к созданию неточных описаний. Часто лучше быть конкретным и прямым. Аналогия здесь очень похожа на эффективную коммуникацию - чем прямее, тем эффективнее передается сообщение.

Например, вы можете заинтересоваться изучением концепции промптинга. Вы можете попробовать что-то вроде:

*Explain the concept prompt engineering. Keep the explanation short, only a few sentences, and don't be too descriptive.*

Из промпта выше не ясно, сколько предложений использовать и в каком стиле. Возможно, с помощью таких промптов вы все равно получите неплохие ответы, но лучший промпт будет очень конкретным, кратким и по существу. Например:

*Use 2-3 sentences to explain the concept of prompt engineering to a high school student.*

### ***Делать или не делать?***

Еще один распространенный совет при разработке промптов - избегать формулировки того, что не нужно делать, а вместо этого указывать, что нужно делать. Это способствует большей специфичности и фокусу на деталях, которые приводят к хорошим результатам модели.

Вот пример неудачного чат-бота по рекомендации фильмов, когда я указываю, что не нужно делать, из-за того, как я сформулировал инструкцию - фокусируясь на том, что не нужно делать.

*Промпт:*

*The following is an agent that recommends movies to a customer. DO NOT ASK FOR INTERESTS. DO NOT ASK FOR PERSONAL INFORMATION.*

*Customer: Please recommend a movie based on my interests.*

*Agent:*

*Результат:*

*Sure, I can recommend a movie based on your interests. What kind of movie would you like to watch? Do you prefer action, comedy, romance, or something else?*

Вот лучший промпт:

*Промпт:*

*The following is an agent that recommends movies to a customer. The agent is responsible to recommend a movie from the top global trending movies.*

*It should refrain from asking users for their preferences and avoid asking for personal information. If the agent doesn't have a movie to recommend, it should respond "Sorry, couldn't find a movie to recommend today."*

*Customer: Please recommend a movie based on my interests.*

*Agent:*

*Результат:*

*Sorry, I don't have any information about your interests. However, here's a list of the top global trending movies right now: [list of movies]. I hope you find something you like!*

### **Примеры промптов**

#### ***Сжатие текста***

Одна из стандартных задач в генерации естественного языка - это сжатие текста. Сжатие текста может включать множество различных вариантов и областей. Фактически, одно из наиболее перспективных применений языковых моделей - это возможность сжатия статей и концепций в быстрые и легко читаемые резюме. Давайте попробуем базовую задачу сжатия текста с помощью промптов.

Предположим, вы хотите узнать о антибиотиках, и вы можете попробовать следующий промпт:

*Промпт:*

*Explain antibiotics*

*A:*

*Результат:*

*Antibiotics are a type of medication used to treat bacterial infections. They work by either killing the bacteria or preventing them from reproducing, allowing the body's immune system to fight off the infection. Antibiotics are usually taken orally in the form of pills, capsules, or liquid solutions, or sometimes administered intravenously. They are not effective against viral infections, and using them inappropriately can lead to antibiotic resistance.*

В формате "A:" указывается явный промпт для ответа, который вы используете в вопросно-ответной форме. Вы использовали это здесь, чтобы сообщить модели, что ожидается продолжение. В этом примере не совсем ясно, насколько это помогает, но мы вернемся к этому позже. Предположим, что это слишком много информации, и вам хотелось бы сжать ее еще больше. На самом деле, вы можете указать модели сжать ее в одно предложение, например, так:

*Промпт:*

*Antibiotics are a type of medication used to treat bacterial infections. They work by either killing the bacteria or preventing them from reproducing, allowing the body's immune system to fight off the infection. Antibiotics are usually taken orally in the form of pills, capsules, or liquid solutions, or sometimes administered intravenously. They are not effective against viral infections, and using them inappropriately can lead to antibiotic resistance.*

*Explain the above in one sentence:*

*Результат:*

*Antibiotics are medications used to treat bacterial infections by either killing the bacteria or stopping them from reproducing, but they are not effective against viruses and overuse can lead to antibiotic resistance.*

Не обращая слишком много внимания на точность вывода выше, о которой мы поговорим позже, модель попыталась сжать параграф в одно предложение. Вы можете экспериментировать с инструкциями, чтобы улучшить результаты.

### ***Извлечение информации***

Хотя языковые модели обучаются выполнять генерацию естественного языка и связанные задачи, они также способны выполнять классификацию и ряд других задач обработки естественного языка (NLP).

Вот пример промпта, извлекающего информацию из заданного параграфа.

*Промпт:*

*Author-contribution statements and acknowledgements in research papers should state clearly and specifically whether, and to what extent, the authors used AI technologies such as ChatGPT in the preparation of their manuscript and analysis. They should also indicate which LLMs were used. This will alert editors and reviewers to scrutinize manuscripts more carefully for potential biases, inaccuracies and improper source crediting. Likewise, scientific journals should be transparent about their use of LLMs, for example when selecting submitted manuscripts.*

*Mention the large language model based product mentioned in the paragraph above:*

*Результат:*

*The large language model based product mentioned in the paragraph above is ChatGPT.*

Есть множество способов улучшить результаты выше, но это уже очень полезно.

К настоящему моменту должно быть очевидно, что вы можете попросить модель выполнить различные задачи, просто указав ей, что делать. Это мощная возможность, которую уже используют разработчики продуктов и искусственного интеллекта для опытов и создания полезных приложений.

### ***Вопросно-ответные системы (Q&A)***

Один из лучших способов получить от модели конкретные ответы - это улучшить формат промпта. Как было рассмотрено ранее, промпт может сочетать в себе инструкции, контекст, входные данные и индикаторы вывода для получения улучшенных результатов. Хотя эти компоненты не являются обязательными, это является хорошей практикой, поскольку чем более конкретными вы будете в инструкции, тем лучше результаты вы получите. Ниже приведен пример того, как это может выглядеть при использовании более структурированного промпта.

*Промпт:*

*Answer the question based on the context below. Keep the answer short and concise. Respond "Unsure about answer" if not sure about the answer.*

*Context: Teplizumab traces its roots to a New Jersey drug company called Ortho Pharmaceutical. There, scientists generated an early version of the antibody, dubbed OKT3. Originally sourced from mice, the molecule was able to bind to the surface of T cells and limit their cell-killing potential. In 1986, it was approved to help prevent organ rejection after kidney transplants, making it the first therapeutic antibody allowed for human use.*

*Question: What was OKT3 originally sourced from?*

*Answer:*

*Результат:*

*Mice.*

### ***Классификация текста***

До сих пор вы использовали простые инструкции для выполнения задачи. В качестве промпт-инженера вам нужно стать лучше в предоставлении более точных инструкций. Но это еще не все! Вы также обнаружите, что для более сложных случаев просто предоставление инструкций будет недостаточно. Здесь вам нужно подумать больше о контексте и разных элементах, которые можно использовать в промпте. Другие элементы, которые вы можете предоставить, это входные данные или примеры.

Давайте попробуем продемонстрировать это на примере классификации текста.

*Промпт:*

*Classify the text into neutral, negative or positive.*

*Text: I think the food was okay.*

*Sentiment:*

*Результат:*

*Neutral*

Вы дали инструкцию классифицировать текст, и модель вернула 'Neutral', что является правильным. В этом ничего плохого, но допустим, что вам действительно нужно, чтобы модель возвращала метку в точно нужном формате. Вместо Neutral вы хотите, чтобы она возвращала neutral. Как это достичь? Есть несколько способов сделать это. Вам важна конкретность, поэтому чем больше информации вы можете предоставить промпту, тем лучше результаты. Вы можете попробовать предоставить примеры, чтобы указать правильное поведение. Давайте попробуем снова:

*Промпт:*

*Classify the text into neutral, negative or positive.*

*Text: I think the vacation is okay.*

*Sentiment: neutral*

*Text: I think the food was okay.*

*Sentiment:*

*Результат:*

*neutral*

Отлично! На этот раз модель вернула neutral, что является точной меткой, которую вы искали. Кажется, что пример, предоставленный в промпте, помог модели быть конкретной в выводе.

Чтобы показать, почему иногда важна конкретность, посмотрите на следующий пример и найдите проблему:

*Промпт:*

*Classify the text into nutral, negative or positive.*

*Text: I think the vacation is okay.*

*Sentiment:*

*Результат:*

*Neutral*

В чем проблема? На подсказку, придуманная метка nutral полностью игнорируется моделью. Вместо этого модель выводит Нейтральный, так

как она немного предвзята к этой метке. Но предположим, что вам действительно нужно *neutral*. Как исправить это? Возможно, вы можете попробовать добавить описания к меткам или добавить больше примеров в промпт? Если вы не уверены, мы обсудим несколько идей в следующих разделах.

### *Диалог*

Одно из самых интересных применений промпт-инженерии - это инструктирование LLM-системы о том, как вести себя, какая должна быть ее намеренность и какая должна быть ее личность. Это особенно полезно, когда вы создаете системы диалога, такие как чат-боты для обслуживания клиентов.

Например, давайте создадим систему диалога, которая способна генерировать более технические и научные ответы на вопросы. Обратите внимание, как вы явно указываете, как она должна себя вести через инструкцию. Иногда это называют ролевой промптинг.

#### *Промпт:*

*The following is a conversation with an AI research assistant. The assistant tone is technical and scientific.*

*Human: Hello, who are you?*

*AI: Greeting! I am an AI research assistant. How can I help you today?*

*Human: Can you tell me about the creation of blackholes?*

*AI:*

#### *Результат:*

*Sure! Black holes are regions of spacetime where the gravitational force is so strong that nothing, not even light, can escape from it. They are created when a very massive star dies and its core collapses in on itself, forming a singularity of infinite density. The intense gravity of the singularity pulls in all the matter and radiation around it, creating the black hole.*

Наш исследовательский ассистент звучит слишком технически, верно? Хорошо, давайте изменим это поведение и научим систему давать более доступные ответы.

#### *Промпт:*

*The following is a conversation with an AI research assistant. The assistant answers should be easy to understand even by primary school students.*

*Human: Hello, who are you?*

*AI: Greeting! I am an AI research assistant. How can I help you today?*

*Human: Can you tell me about the creation of black holes?*

*AI:*

*Результат:*

*Sure! Black holes are created when a very massive star runs out of fuel and collapses in on itself. The gravity of the star is so strong that nothing, not even light, can escape its pull. This creates a point in space-time with an intense gravitational force called a black hole.*

Вы сделали некоторый прогресс. Можно продолжать улучшать это. Если вы добавите больше примеров, вы можете получить еще лучшие результаты.

### ***Генерация кода***

Одно из применений, в которых LLMs довольно эффективны, - это генерация кода. Copilot - прекрасный пример этого. С помощью хорошо продуманных промптов вы можете выполнять множество задач по генерации кода. Рассмотрим несколько примеров ниже.

Сначала попробуем простую программу, которая приветствует пользователя.

*Промпт:*

```
/*  
Ask the user for their name and say "Hello"  
*/
```

*Результат:*

```
let name = prompt("What is your name?");  
console.log(`Hello, ${name}!`);
```

Обратите внимание, что вам даже не нужно указывать язык программирования.

Давайте немного повысим уровень. Приведенный ниже пример показывает, насколько мощными могут быть LLMs с небольшими усилиями в разработке промптов.

*Промпт:*

```
""""  
Table departments, columns = [DepartmentId, DepartmentName]  
Table students, columns = [DepartmentId, StudentId, StudentName]  
Create a MySQL query for all students in the Computer Science  
Department  
""""
```

*Результат:*

```
SELECT StudentId, StudentName
FROM students
WHERE DepartmentId IN (SELECT DepartmentId FROM departments
WHERE DepartmentName = 'Computer Science');
```

Это очень впечатляюще. В этом примере вы предоставили данные о схеме базы данных и попросили его сгенерировать корректный запрос MySQL.

### **Рассуждение**

Возможно, одна из самых сложных задач для LLM на сегодняшний день - это задача, требующая некоторой формы рассуждения. Рассуждение является одной из наиболее интересных областей из-за типов сложных приложений, которые могут возникнуть из LLM.

Были достигнуты некоторые успехи в задачах, связанных с математическими возможностями. Однако важно отметить, что текущие LLM все еще испытывают трудности с выполнением задач рассуждения, поэтому для этого требуются еще более продвинутые техники промпт-инженерии. Пока что мы рассмотрим несколько базовых примеров для показа математических возможностей.

*Промпт:*

*What is 9,000 \* 9,000?*

*Результат:*

*81,000,000*

*Давайте попробуем что-то более сложное.*

*Промпт:*

*The odd numbers in this group add up to an even number: 15, 32, 5, 13, 82, 7, 1.*

*A:*

*Результат:*

*No, the odd numbers in this group add up to an odd number: 119.*

*Ответ неправильный! Попробуем улучшить это, улучшив промпт.*

*Промпт:*

*The odd numbers in this group add up to an even number: 15, 32, 5, 13, 82, 7, 1.*

*Solve by breaking the problem into steps. First, identify the odd numbers, add them, and indicate whether the result is odd or even.*

*Результат:*

*Odd numbers: 15, 5, 13, 7, 1*

*Sum: 41*

*41 is an odd number.*

## **Создание цепочек промптов**

### ***Введение в Создание цепочек промптов***

Для повышения надежности и производительности LLM, одним из важных приемов промпт-инжиниринга является разбиение задачи на подзадачи. После того как эти подзадачи определены, для LLM предлагаются подзадачи, а затем ее ответ используется в качестве входных данных для другого запроса. Это то, что называется Созданием цепочек промптов, когда задача разбивается на подзадачи с целью создания цепочки операций промптов.

Создание цепочек промптов полезно для решения сложных задач, которые LLM может с трудом решить, если будет предложен очень большой промпт. При создании цепочки промптов, цепочки промптов выполняют преобразования или дополнительные процессы над сгенерированными ответами, прежде чем достичь конечного желаемой формы.

Помимо повышения производительности, цепочки промптов помогают повысить прозрачность вашего применения LLM, повышает управляемость и надежность. Это означает, что вы можете гораздо проще отлаживать проблемы с ответами модели, а также анализировать и улучшать производительность на различных этапах, которые нуждаются в улучшении.

Цепочка промптов особенно полезна при создании диалоговых помощников на базе LLM и улучшении персонализации и пользовательского опыта ваших приложений.

### ***Случаи использования Создания цепочек промптов***

#### ***Создание цепочек промптов для контроля качества документа***

Цепочки промптов могут использоваться в различных сценариях, которые могут включать несколько операций или преобразований. Например, один из распространенных вариантов использования LLM предполагает ответы на вопросы о большом текстовом документе.

Вам поможет, если вы создадите два разных промпта, где первый промпт отвечает за извлечение соответствующих цитат для ответа на вопрос, а второй промпт принимает в качестве входных данных цитаты и исходный документ для ответа на заданный вопрос. Другими словами, вы

создадите две разных промпта для выполнения ответа на вопрос о документе.

Первый промпт ниже извлекает соответствующие цитаты из документа, заданного в вопросе. Обратите внимание, что для простоты мы добавили заполнитель для документа `{{document}}`. Чтобы протестировать подсказку, вы можете скопировать и вставить статью из Википедии, например, эту страницу, посвященную технике подсказок. Из-за более большого контекста, используемого для этой задачи, мы используем модель `gpt-4-1106-preview` от OpenAI. Вы можете использовать промты с длинным контекстом с другими LLM, такими как Claude.

*Prompt 1:*

*You are a helpful assistant. Your task is to help answer a question given a document. The first step is to extract quotes relevant to the question from the document, delimited by #####. Please output the list of quotes using `<quotes>``</quotes>`. Respond with "No relevant quotes found!" if no relevant quotes were found.*

*#####*

*{{document}}*

*#####*

Вот скриншот всего промпта, включая вопрос, который передается с использованием роли пользователя (рисунок 20).

*Output of Prompt 1:*

*`<quotes>`*

- Chain-of-thought (CoT) prompting[27]*
- Generated knowledge prompting[37]*
- Least-to-most prompting[38]*
- Self-consistency decoding[39]*
- Complexity-based prompting[41]*
- Self-refine[42]*
- Tree-of-thought prompting[43]*
- Maieutic prompting[45]*
- Directional-stimulus prompting[46]*
- Textual inversion and embeddings[59]*
- Using gradient descent to search for prompts[61][62][63][64]*
- Prompt injection[65][66][67]*

*`</quotes>`*

Цитаты, возвращенные в первом промпте, теперь можно использовать в качестве входных данных для второго промпта ниже. Обратите внимание, что вы можете еще немного очистить цитаты, то есть удалить цитаты. Эти цитаты можно удалить или использовать как часть другой подсказки в цепочке, но пока мы это проигнорируем.

The screenshot shows the 'Playground' interface with a chat window on the left and model settings on the right.

**Chat Window:**

- SYSTEM:** You are a helpful assistant. Your task is to help answer a question given in a document. The first step is to extract quotes relevant to the question from the document, delimited by ####. Please output the list of quotes using <quotes></quotes>. Respond with "No relevant quotes found!" if no relevant quotes were found.
- USER:** Question: What are the prompting techniques mentioned in the document?
- SYSTEM (continued):** #### Prompt engineering is the process of structuring text that can be interpreted and understood by a generative AI model.[1][2] A prompt is natural language text describing the task that an AI should perform.[3]

**Model Settings:**

- Model: gpt-4-1106-preview
- Temperature: 1
- Maximum length: 256
- Stop sequences: Enter sequence and press Tab
- Top P: 1
- Frequency penalty: 0
- Presence penalty: 0

API and Playground requests will not be used to train our models. [Learn more](#)

Рис. 20. Пример промпта на извлечение цитат из текста

Как видите, упрощение и создание цепочек промптов — это полезный подход к промптам, когда ответы необходимо подвергнуть нескольким операциям или преобразованиям.

### Основная структура промпта

Действуй в качестве [РОЛЬ], выполни [ЗАДАЧА] в [ФОРМАТ]

|                      |                        |                |
|----------------------|------------------------|----------------|
| <b>ДЕЙСТВУЙ КАК:</b> | <b>ВЫПОЛНИ ЗАДАЧУ:</b> | <b>В ВИДЕ:</b> |
| Маркетолог           | Заголовок              | Таблица        |
| Рекламодатель        | Статья                 | Список         |
| Психолог             | Эссе                   | Сводка         |
| Автор бестселлеров   | Конспект книги         | HTML код       |

|                     |                    |                     |
|---------------------|--------------------|---------------------|
| Терапевт            | Email письмо       | Электронная таблица |
| Дизайнер сайтов     | Пост в соц. сетях  | Графики             |
| Журналист           | Описание продукта  | CSV-файл            |
| Изобретатель        | Письмо             | Текстовый файл      |
| Финансовый директор | Заметка в блоге    | JSON                |
| Копирайтер          | Ключевые слова SEO | Насыщенный текст    |
| Оперативный инженер | Резюме             | PDF                 |
| Бухгалтер           | Видеосценарий      | XML                 |
| Юрист               | Рецепт             | Markdown            |
| Аналитик            | Продающий текст    | Диаграмма Ганта     |
| Менеджер проектов   | Анализ             | Облако слов         |

#### СВЯЗАННЫЕ ПРОМПТЫ:

- 1-Предоставь мне идеальный набросок для эффективной и убедительной статьи в блоге.
- 2-Напиши список привлекающих внимание заголовков для этой записи в блоге на основе [тема].
- 3- Напиши список слов подзаголовка и хуков для этой же записи в блоге.
- 4-Напиши список ключевых слов для этого блога
- 5-Напиши список убедительных призывов к действию для этой записи блога.
- 6-Объедини лучший заголовок с лучшими подзаголовками, хуками, ключевыми словами и призывами к действию, чтобы написать пост в блоге на [тему].
- 7-Перепиши этот пост в [стиль], [тон], [голос], и личность.

#### СЛОИ В ПРОМПТАХ:

- ZERO – Напиши мне 5 заголовков о [теме].
- SINGLE – Напиши мне 5 заголовков о [теме]. Вот пример одного заголовка: 5 способов сбросить вес.
- MULTIPLE – Напиши мне 5 заголовков о [теме]. Вот пример нескольких заголовков: 5 способов сбросить вес, как сбросить больше жира за 4 недели, попрощайтесь с упрямым жиром. Найдите более быстрый способ сбросить вес.

#### МАРКЕТИНГОВЫЕ ПРОМПТЫ:

- Дай мне список недорогих идей, как лучше продвигать свой бизнес.
- Выступи в роли бизнес-консультанта, каков наилучший способ решения этой проблемы [Проблема].
- Создай 30-дневную стратегию контента для социальных сетей на основе [Тема 1] и [Тема 2].

## Лекция 6. Разработка промптов для сферы культуры

Пользователи активно используют нейросети в профессиональных и развлекательных целях для генерации текстов, изображений, видео и аудио.

Российские разработки успешно конкурируют с иностранными, занимая свою нишу в некоторых категориях:

### 1. Генераторы текста:

GigaChat (Россия),  
YandexGPT (Россия),  
Gerwin (Россия),  
Порфирьевич (Россия),  
ChatGPT,  
Grok,  
Character AI,  
Google Bard,  
Janitor AI,  
Perplexity AI,  
Mistral,  
Mixtral  
<https://quasi.market>  
<https://writewithlaika.com>  
<https://smartscribe.app>  
<https://nichesss.com>  
<https://compose.ai>  
<https://text-gen.com>  
<https://webcopilot.co>  
<https://frase.io>  
<https://affiliate.notion.so>  
<https://languagetool.org>  
<https://aiduh.com>  
<https://writelyai.com>  
<https://app.grammarly.com>  
<https://proposalgenie.ai>  
<https://chrome.google.com>  
<https://luciaai.com>  
<https://caliberai.net>  
<https://helloscribe.ai>

<https://wordai.com>  
<https://wordtune.com>  
<https://redacta.me>  
<https://othersideai.com>  
<https://glasp.co>  
<https://paragraphai.com>  
<https://dream.page>  
<https://maester.app>  
<https://elephas.app>  
<https://sudowrite.com>  
<https://lex.page>  
<https://handyplugins.co>  
<https://penelope-ai.vercel.app>

2. *Генераторы иллюстраций:*

Kandinsky (Россия),  
Fusion Brain (Россия),  
Шедеврум (Россия),  
Gerwin,  
MidJourney,  
Stable Diffusion,  
Dall-E,  
Leonardo AI  
<https://stablehorde.net>  
<https://gocharlie.ai>  
<https://stability.ai>  
<https://getimg.ai>  
<https://aragon.ai>  
<https://rocketai.io>  
<https://pollinations.ai>  
<https://diffusion.land>  
<https://openai.com>  
<https://getalpaca.io>  
<https://canva.com>  
<https://stockai.com>  
<https://craiyon.com>  
<https://soreal.ai>  
<https://imgcreator.zmo.ai>

<https://stylized.ai>  
<https://artssy.co>  
<https://nijijourney.com>  
<https://roll-art-die.com>  
<https://slashdreamer.com>  
<https://enterpix.app>  
<https://xno.ai>

3. *Генераторы видео:*

Visper (Россия),  
Sora,  
CapCut  
<https://replicate.com>  
<https://astria.ai>  
<https://waymark.com>  
<https://flike.ai>  
<https://synthesia.io>  
<https://steve.ai>  
<https://invideo.io>  
<https://opus.ai>  
<https://wowto.ai>  
<https://colossyan.com>  
<https://xpressioncamera.com>  
<https://movio.la>  
<https://pyttipanna.xyz>  
<https://peech-ai.com>  
<https://wonderdynamics.com>  
<https://studio.d-id.com>  
<https://livereacting.com>  
<https://audiolabs.io>  
<https://hourone.ai>

4. *Генераторы аудио:*

Маэстро (Россия),  
ElevenLabs,  
Suno AI  
<https://naturallanguageplaylist.com>  
<https://endel.io>  
<https://harmonai.org>

<https://riffusion.com>  
<https://sonify.io>  
<https://beatoven.ai>  
<https://ampermusic.com>  
<https://soundful.com>  
<https://songtell.com>  
<https://sweetcocoa.github.io>  
<https://boomy.com>  
<https://openvoiceos.com>  
<https://audialab.com>  
<https://mubert.com>  
<https://splashmusic.com>  
<https://aiva.ai>

**Сервисы, полезные для развития креативности:**

1. [innovationhack.io](https://innovationhack.io) — очень свежий проект, позволяющий проводить исследования, генерировать продуктовые идеи, создавать стратегии и инновации. Есть большая база готовых темплейтов под разные задачи и под капотом этого сервиса ИИ.

2. [ai.boardofinnovation.com](https://ai.boardofinnovation.com) — Board of Innovation одни из самых классных инноваторов в мире, на мой взгляд. Не такие матёрые как IDEO, но всё-равно крутые. Сделали свой бесплатный ИИ-сервис для генерации идей, разработки продуктов, работы с исследованиями, персонами, сценариями.

3. [thebrain.com](https://thebrain.com) — Очень старый проект, который позволяет создать «свой цифровой мозг». Не остался в стороне от развития ИИ, что очень хорошо. Позиционируют себя как идеальную платформу для мышления и базу знаний «без ограничений». Это похоже на сервис интеллект-карты, но в 3D.

4. [inspiq.com](https://inspiq.com) — ещё один свежий сервис на основе ИИ, помогающий придумывать. От единственной мысли к полноценному плану запуска за считанные секунды.

5. [ideator.design](https://ideator.design) — сервис на основе ИИ, созданный UX-дизайнером из JP Morgan London. Помогает продуктовым и дизайн-командам быстро создавать описания концепций дизайна и интерактивных систем, пользовательских потоков, сценарных ограничений, POV и других важных элементов UX/UI.

6. ideamap.ai — центр мозговых штурмов, управляемый искусственным интеллектом. Практически бесконечная доска для рисования интеллект-карты с использованием разных механик генерации идей.

7. uniqium.com — инструмент для мозгового штурма на основе искусственного интеллекта, предназначенный для повышения эффективности мозговых штурмов.

8. stormz.me — сервис для проведения мозговых штурмов, работающий на основе искусственного интеллекта. Структурирован внутри по этапам, содержит подсказки и неплохо выглядит.

9. ideagenerator.ai — генератор идей на базе искусственного интеллекта, помогает придумывать и развивать идеи. Очень простой и доступный. И автор написал, что он будет бесплатным всегда!

10. ideator.mit.edu — сервис от MIT с искусственным интеллектом, который поможет придумывать идеи и находить решения многих проблем. Чтобы попробовать, надо отправить запрос в лист ожидания. Мне подтвердили через пару дней.

Еще больше сервисов на основе искусственного интеллекта собраны в этом списке (рисунок 21):

<https://docs.google.com/spreadsheets/d/1sSaVgVlgktBPlK5V5Mozvq4DI4pMiMfp1Ubgg17bieI/edit?usp=sharing>



Рис. 21 Список нейросетей

### 3. ПРАКТИЧЕСКИЙ РАЗДЕЛ

#### 3.1 ОПИСАНИЕ ЛАБОРАТОРНЫХ РАБОТ

**Тема 5. Информационные продукты на основе искусственного интеллекта для решения практических задач в сфере культуры**

**Лабораторная работа №1**  
**Создание с помощью ИИ программного кода**  
**для макросов Google Workspace (4 часа)**

**Задание 1.**

- Ознакомиться с расширением Apps Scripts для Google Workspace
- Для документа Google Docs создать с помощью ИИ программный код, вставляющий в него таблицу конкретной структуры с заданной шапкой. Добавить кнопку вставки таблицы на панель Google Docs текущего документа.

**Задание 2.**

- Ознакомиться с основными задачами и возможностями ключа программного интерфейса приложения (API)
- Для документа Google Spreadsheets создать с помощью ИИ программный код, берущий длинные адреса интернет-источников из первого столбца, сокращающий их с помощью выбранного интернет-сервиса и помещающий во второй столбец напротив. Добавить кнопку вставки на панель Google Spreadsheets текущего документа.

**Лабораторная работа №2**

**ИИ-ассистенты для культурных туров и экскурсий (4 часа)**

**Задание 1**

- Разработать ИИ-ассистент для виртуальных туров и экскурсий.

**Задание 2**

- Создать голосовой ассистент, способный проводить виртуальные туры и экскурсии.

## **Лабораторная работа №2**

### **Программирование чат-ботов с разным функционалом интерактивного взаимодействия с пользователем (6 часов)**

#### **Задание 1**

- Разработать схему интерактивного взаимодействия чат-бота для консультаций по вопросам культурного наследия и пользователя, с фиксацией инструментария реализации такого взаимодействия.

### **Список заданий для самоподготовки**

#### **Тема 5. Информационные продукты на основе искусственного интеллекта для решения практических задач в сфере культуры**

1. Разработка сервиса для анализа культурного контента и выявления его ключевых характеристик

Описание: Создание и обучение модели ИИ для анализа текстов, изображений и видеоматериалов с целью выделения ключевых тем и характеристик. Оценка точности и применимости разработанного сервиса.

2. Программирование ИИ-ассистентов и гидов для культурных туров и экскурсий

Описание: Разработка чат-ботов или голосовых ассистентов, способных проводить виртуальные туры и экскурсии. Обучение моделей на основе информации о культурных объектах и создание интерактивного опыта для пользователей.

3. Создание инструментов для цифровой каталогизации и архивирования культурного наследия

Описание: Программирование системы для автоматического сканирования, распознавания и классификации культурных артефактов. Обеспечение точности и удобства использования системы для специалистов по культурному наследию.

4. Создание инструментов для виртуального обучения и обмена культурным опытом

Описание: Разработка платформы для дистанционного обучения и виртуальных мастер-классов с использованием ИИ. Интеграция функций для взаимодействия пользователей и обмена культурным опытом.

5. Разработка инструментов для автоматической обработки и адаптации культурного контента под различные платформы и форматы

Описание: Создание алгоритмов для адаптации текстов, изображений и видеоматериалов под требования различных платформ (социальные сети, веб-сайты, мобильные приложения). Тестирование на различных типах контента и платформ.

6. Программирование систем рекомендаций и персонализации культурных предложений

Описание: Разработка системы рекомендаций культурных мероприятий, книг, фильмов и других форм искусства на основе предпочтений пользователей. Использование данных для улучшения точности рекомендаций.

7. Разработка сервиса для автоматического создания афиш и рекламных материалов для культурных событий

Описание: Программирование ИИ-инструмента для генерации афиш, постеров и рекламных материалов на основе описания события. Настройка шаблонов и тестирование качества визуальных материалов.

8. Создание системы для анализа и интерпретации отзывов о культурных мероприятиях

Описание: Разработка алгоритмов для анализа отзывов и комментариев пользователей о культурных событиях. Оценка настроения и выделение ключевых тем для улучшения качества мероприятий.

9. Программирование сервиса для автоматического создания сценариев культурных мероприятий

Описание: Разработка ИИ, способного генерировать сценарии для различных культурных мероприятий (концерты, театральные постановки, фестивали). Тестирование на основе реальных событий.

10. Создание системы для мониторинга и прогнозирования трендов в культуре

Описание: Разработка инструмента для анализа данных из социальных сетей, новостей и других источников для выявления трендов в культурной сфере. Прогнозирование будущих трендов на основе исторических данных.

11. Разработка чат-бота для консультаций по вопросам культурного наследия

Описание: Создание чат-бота, который может отвечать на вопросы пользователей о культурных объектах и событиях. Обучение модели на базе данных о культурных достопримечательностях и истории.

12. Создание инструмента для анализа и оптимизации расписания культурных мероприятий

Описание: Программирование ИИ-инструмента для анализа расписания культурных мероприятий и оптимизации времени проведения для максимального привлечения аудитории. Тестирование и оценка эффективности.

## **3.2 ОПИСАНИЕ ПРАКТИЧЕСКИХ РАБОТ**

### **Тема 2. Этика использования ИИ и его влияние на сферу культуры**

#### **Практическая работа 1.**

##### **Этические вызовы и производственные преимущества внедрения ИИ в сферу культуры (4 часа)**

###### **Задание 1**

- Рассмотреть этические аспекты и вызовы применения ИИ в культуре.
- Провести дискурс на тему этических вопросов, связанных с использованием ИИ в создании и распространении культурного контента, включая авторские права и влияние на творчество.

###### **Задание 2:**

- Рассмотреть социальные и культурные последствия внедрения ИИ.
- Проанализировать влияния ИИ на культурное разнообразие, доступность культуры и изменения в профессиональной сфере искусств.

### **Тема 3. Промт-инжиниринг: понятие и основные принципы**

#### **Практическая работа 1.**

##### **Основные понятия и концепции в промт-инжиниринге (2 часа)**

###### **Задания:**

- Рассмотреть Основные понятия и концепции в промт-инжиниринге.
- Провести анализ существующих подходов в промт-инжиниринге.
- Рассмотреть вопросы интеграции промт-инжиниринга в рабочие процессы культурной индустрии.

#### **Практическая работа 2.**

##### **Промт-инжиниринг в литературном творчестве (4 часа)**

###### **Задание1**

- Выявить особенности промт-инжиниринга в литературном творчестве. Создать примеры поэтического творчества.

###### **Задание 2**

- Выявить особенности промпт-инжиниринга в написании экспертных статей. Написать экспертную на выбранную тему сферы культуры, используя соответствующие методы промпт-инжиниринга. Написанную статью подкрепить релевантными источниками по теме. Для нахождения источников использовать специальные нейронные сети для поддержки научной деятельности.

#### **Тема 4. Разработка промтов для сферы культуры**

##### **Практическая работа 1.**

##### **Промпт-инжиниринг для генерации визуального искусства (2 часа)**

###### **Задания:**

- Рассмотреть вопросы применения промпт-инжиниринга для генерации визуального искусства.
- Используя технологии промпт-инжиниринга создать поздравительную открытку с ответствующим изображением и надписью к выбранному празднику.
- Используя технологии промпт-инжиниринга создать приглашение на мероприятие.

##### **Практическая работа 2.**

##### **Промпт-инжиниринг в обработке аудиоинформации (2)**

###### **Задания:**

- Рассмотреть вопросы применения промпт-инжиниринга в сфере обработки аудио информации.
- Рассмотреть вопросы применения промпт-инжиниринга в музыкальном творчестве.
- Используя технологии промпт-инжиниринга написать текст, освещающий информацию о музейном экспонате и озвучить ее с помощью выбранных неросетей.
- Используя технологии промпт-инжиниринга сгенерировать музыкальный ряд для фона подачи информации о музейном экспонате.
- Объединить аудио с текстом о музейном экспонате и музыкальный аудиоряд.

### **Практическая работа 3.**

#### **Примеры успешного применения промпт-инжиниринга в культурных проектах (4)**

##### **Задания:**

- Рассмотреть вопросы применения промпт-инжиниринга в сфере обработки видео информации.
- Рассмотреть вопросы применения промпт-инжиниринга для создания рекламных роликов.
- Используя технологии промпт-инжиниринга написать сценарий для рекламного видеоролика, освещающего выбранное культурное событие/мероприятие.
- Используя технологии промпт-инжиниринга сгенерировать видеоряд для рекламного видеоролика, освещающего выбранное культурное событие/мероприятие.
- Используя технологии промпт-инжиниринга написать текст для рекламного видеоролика, освещающего выбранное культурное событие/мероприятие и озвучить его.
- Используя технологии промпт-инжиниринга сгенерировать музыкальный ряд для фона подачи информации в видеоролике.
- Объединить видеоряд, аудио с текстом и музыкальный аудио-ряд создав таким образом видеоролик, освещающий выбранное культурное событие/мероприятие.

#### **Список вопросов для самоподготовки**

##### ***Тема 1. Понятие ИИ и языковых моделей***

1. Основные подходы и методы в разработке языковых моделей ИИ  
Обсуждение различных методов и алгоритмов, используемых в создании языковых моделей ИИ, включая статистические модели и нейронные сети.
2. Применение языковых моделей в культурной индустрии  
Исследование примеров использования языковых моделей для создания текстов, перевода, анализа данных и других задач в культурной сфере.

## **Тема 2. Этика использования ИИ и его влияние на сферу культуры**

### **1. Этические аспекты и вызовы применения ИИ в культуре**

Обсуждение этических вопросов, связанных с использованием ИИ в создании и распространении культурного контента, включая авторские права и влияние на творчество.

### **2. Социальные и культурные последствия внедрения ИИ**

Анализ влияния ИИ на культурное разнообразие, доступность культуры и изменения в профессиональной сфере искусств.

## **Тема 3. Промт-инжиниринг: понятие и основные принципы**

### **1. Теоретические основы и методы промт-инжиниринга**

2. Обсуждение концептуальных основ промт-инжиниринга, различных подходов и методов для создания эффективных промтов.

3. Практические примеры и кейсы использования промт-инжиниринга

4. Разбор конкретных случаев успешного применения промт-инжиниринга в различных проектах, анализ эффективности и результатов.

5. Примеры успешного применения промт-инжиниринга в культурных проектах.

6. Перспективы использования промт-инжиниринга для инноваций в культурной сфере.

7. Создание промтов для интерактивных и виртуальных культурных мероприятий.

8. Разработка промтов для сценариев и игровых проектов

9. Обучение специалистов по промт-инжинирингу для культурных целей.

10. Оценка и корректировка эффективности промтов для различных жанров и форматов культурного контента.

## **Тема 4. Разработка промтов для сферы культуры**

1. Создание промтов для генерации графических изображений и видеоматериалов.

2. Разработка промптов для создания литературных произведений с использованием ИИ.

3. Использование промптов для разработки сценариев и сценических произведений.

4. Промпт-инжиниринг для создания музыкальных композиций и звукового дизайна.

5. Разработка промптов для создания культурно-исторических реконструкций и виртуальных туров.

6. Промпты для создания текстов культурных мероприятий, рекламных материалов и программных буклетов.

7. Создание промптов для генерации графических изображений и видеоматериалов.

Описание: Разработка и тестирование промптов для генерации изображений и видеоматериалов с использованием различных ИИ-инструментов, таких как DALL-E или Midjourney. Оценка качества сгенерированных материалов и их соответствие заданной тематике.

8. Разработка промптов для создания литературных произведений с использованием ИИ.

Описание: Написание промптов для генерации стихотворений, рассказов или эссе с использованием ИИ, таких как GPT. Анализ и корректировка сгенерированных текстов для достижения литературного качества и тематической глубины.

9. Использование промптов для разработки сценариев и сценических произведений.

Описание: Создание промптов для написания театральных сценариев или сценариев для короткометражных фильмов с помощью ИИ. Работа над структурой сценариев, диалогами и сценическими инструкциями.

10. Промпт-инжиниринг для создания музыкальных композиций и звукового дизайна.

Описание: Разработка промптов для генерации музыкальных композиций и звукового дизайна с помощью ИИ-инструментов, таких как Amper Music или AIVA. Оценка музыкальных произведений на соответствие заданному жанру и настроению.

11. Разработка промптов для создания культурно-исторических реконструкций и виртуальных туров.

Описание: Создание промптов для генерации текстов и визуальных материалов для культурно-исторических реконструкций и виртуальных туров. Работа над детализацией и достоверностью исторических данных.

12. Использование промптов для создания интерактивных инсталляций и выставочных проектов.

Описание: Разработка промптов для генерации контента для интерактивных инсталляций и выставок. Тестирование и адаптация сгенерированных материалов для использования в реальных проектах.

13. Промпт-инжиниринг для создания интерактивных искусственных персонажей и игровых сценариев.

Описание: Создание промптов для разработки искусственных персонажей и игровых сценариев с помощью ИИ. Анализ и корректировка диалогов и действий персонажей для создания увлекательного игрового процесса.

14. Промпты для создания текстов культурных мероприятий, рекламных материалов и программных буклетов.

Описание: Разработка промптов для генерации текстов для программных буклетов, рекламных материалов и описаний культурных мероприятий. Оценка и редактирование сгенерированных текстов для повышения их привлекательности и информативности.

## **4. РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ**

### **4.1 Задания для контролируемой самостоятельной работы студентов**

#### **1. Анализ эволюции языковых моделей**

Исследуйте и сравните ключевые этапы развития языковых моделей, начиная от классических алгоритмов до современных методов глубокого обучения. Подготовьте презентацию, иллюстрирующую основные достижения и их применение в культурной индустрии.

#### **2. Разработка промптов для литературных произведений**

Создайте промпты для генерации стихотворения или короткого рассказа с использованием ИИ. Опишите процесс разработки промптов и результаты, а также предложите способы их улучшения для повышения качества созданного контента.

#### **3. Этические аспекты использования ИИ в культуре**

Проведите исследование по теме "Этические принципы применения ИИ в культурной сфере". Подготовьте отчет, включающий обзор текущих правовых норм и этических стандартов, а также предложите рекомендации по их совершенствованию.

#### **4. Создание ИИ-ассистента для культурного учреждения**

Разработайте концепцию и прототип ИИ-ассистента для музея или галереи, который сможет проводить виртуальные туры и отвечать на вопросы посетителей. Подготовьте описание функционала и обоснуйте выбор технологий.

#### **5. Примеры использования промпт-инжиниринга в искусстве**

Исследуйте несколько успешных проектов, в которых использовался промпт-инжиниринг для создания произведений искусства. Напишите эссе, анализирующее методы и результаты, а также возможности их применения в других культурных проектах.

#### **6. Создание системы рекомендаций для культурного контента**

Разработайте и протестируйте прототип системы рекомендаций, которая предлагает пользователям культурные мероприятия, книги или фильмы на основе их предпочтений. Подготовьте отчет с описанием алгоритма и анализа его эффективности.

## **4.2 Перечень контрольных вопросов по дисциплине**

- 1) Каковы основные характеристики искусственного интеллекта (ИИ) и как они применяются в современной культурной индустрии?
- 2) Что такое языковые модели (LLM) и какова их роль в системах ИИ?
- 3) Какие методы и алгоритмы используются в разработке языковых моделей ИИ?
- 4) В чем заключается принцип работы нейронных сетей и как они применяются в языковых моделях?
- 5) Какие задачи в культурной сфере можно решить с помощью языковых моделей?
- 6) Каковы основные технологии генерации текста с использованием ИИ и их влияние на культурное творчество?
- 7) Как языковые модели помогают в анализе семантики и контекста в культурных текстах?
- 8) Каковы преимущества и ограничения использования языковых моделей в культурном контексте?
- 9) Какие этические аспекты следует учитывать при использовании ИИ в создании культурных продуктов?
- 10) Какие правовые и интеллектуальные аспекты возникают при использовании ИИ в культурной сфере?
- 11) Как автоматизация культурных процессов с помощью ИИ влияет на социокультурную среду?
- 12) Какие есть примеры успешного применения промпт-инжиниринга в культурной сфере?
- 13) Как промпт-инжиниринг способствует стимулированию творческого процесса в различных областях культуры?
- 14) Какие методы обучения и развития специалистов в области промпт-инжиниринга существуют?
- 15) Какие сервисы на основе ИИ можно разработать для решения практических задач в культурной сфере?

## **4.3 Перечень вопросов для самоподготовки**

### **Тема 1. Понятие ИИ и языковых моделей**

Какие основные методы и алгоритмы используются в разработке языковых моделей ИИ?

Как работают статистические модели и в чем их основные преимущества и недостатки?

Какие нейронные сети применяются для создания языковых моделей и какие задачи они решают лучше всего?

В чем заключается принцип работы трансформеров, таких как GPT, и почему они стали популярными?

Какие примеры использования языковых моделей для создания литературных текстов и сценариев вы можете привести?

Как языковые модели могут помочь в переводе и локализации культурного контента?

Какие инструменты анализа данных на основе ИИ могут быть полезны для исследователей в области культуры?

Какие существуют примеры применения языковых моделей для улучшения взаимодействия с аудиторией в культурной сфере?

## **Тема 2. Этика использования ИИ и его влияние на сферу культуры**

Какие этические вопросы возникают при использовании ИИ для создания культурного контента?

Как использование ИИ влияет на авторские права и интеллектуальную собственность в культурной индустрии?

Какие меры можно принять для обеспечения этического использования ИИ в искусстве и культуре?

В чем заключаются основные вызовы при внедрении ИИ в творческие процессы?

Как использование ИИ может повлиять на культурное разнообразие и сохранение культурного наследия?

Какие последствия имеет внедрение ИИ для доступности культуры и искусства?

Как ИИ меняет профессиональную сферу искусств и какие новые возможности он открывает?

Какие существуют риски и возможные негативные последствия внедрения ИИ в культурную сферу?

## **Тема 3. Промт-инжиниринг: понятие и основные принципы**

Какие теоретические основы лежат в основе промт-инжиниринга и почему они важны?

Какие методы и подходы используются для создания эффективных промтов?

Как можно оценить качество и эффективность разработанных промптов?

Какие существуют успешные примеры применения промпт-инжиниринга в культурных проектах?

Какие результаты были достигнуты с помощью промпт-инжиниринга в этих проектах?

Какие уроки можно извлечь из конкретных кейсов и как они могут быть применены в будущих проектах?

#### **4.4 Перечень вопросов к зачету**

1. Что такое искусственный интеллект и каковы его основные характеристики?
2. Какие существуют основные подходы к разработке языковых моделей искусственного интеллекта?
3. В чем состоит роль языковых моделей в современных системах искусственного интеллекта?
4. Какова эволюция языковых моделей от классических методов до нейронных сетей?
5. Какие задачи можно решать с помощью языковых моделей в культурной сфере?
6. Какие технологии используются для генерации текста с помощью искусственного интеллекта?
7. Какие примеры применения языковых моделей встречаются в культурном творчестве?
8. Какие проблемы и ограничения существуют в развитии языковых моделей для культурных целей?
9. Какие этические аспекты необходимо учитывать при использовании искусственного интеллекта в культурной сфере?
10. Каковы правовые аспекты использования искусственного интеллекта в создании и распространении культурных продуктов?
11. Какие социальные и культурные последствия может вызвать автоматизация культурных процессов с помощью искусственного интеллекта?
12. В чем состоит роль промпт-инжиниринга в культурной индустрии?
13. Какие основные концепции и подходы применяются в промпт-инжиниринге?

14. Какие примеры успешного применения промпт-инжиниринга можно привести в различных областях культуры?
15. Какие задачи можно решать с помощью промпт-инжиниринга в сфере визуального искусства?
16. Как промпт-инжиниринг используется для создания музыкальных композиций и звукового дизайна?
17. Какие инструменты и технологии применяются при разработке промптов для культурно-исторических реконструкций и виртуальных туров?
18. Как промпт-инжиниринг способствует созданию интерактивных инсталляций и выставочных проектов?
19. Как промпт-инжиниринг используется для разработки интерактивных искусственных персонажей и игровых сценариев?
20. Какие примеры можно привести использования промпт-инжиниринга в маркетинге культурных продуктов?
21. Что такое программируемые сервисы на основе искусственного интеллекта и какова их роль в культурной сфере?
22. Какие задачи можно решать с помощью программируемых сервисов в сфере культуры?
23. Каким образом ИИ-ассистенты и гиды используются в культурных турах и экскурсиях?
24. Как программируемые сервисы способствуют цифровой каталогизации и архивированию культурного наследия?
25. Как ИИ поддерживает виртуальное обучение и обмен культурным опытом?
26. Какие примеры успешного применения программируемых сервисов можно назвать в культурной сфере?
27. Какие технологии используются для автоматической обработки и адаптации культурного контента под различные платформы и форматы?
28. Как системы рекомендаций и персонализации культурных предложений работают с использованием данных и ИИ?
29. Какие перспективы развития программируемых сервисов на основе ИИ вы видите в социокультурной деятельности?
30. Какие вызовы и проблемы стоят перед внедрением и развитием ИИ в культурной сфере?

## **5. ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ**

### **5.1 Учебная программа**

#### **СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА**

##### ***Тема 1. Понятие искусственный интеллект и языковые модели***

Определение искусственного интеллекта (ИИ) и его ключевые характеристики. Роль языковых моделей (LLM) в современных системах искусственного интеллекта. Эволюция языковых моделей от классических алгоритмов до глубокого обучения. Принцип работы нейронных сетей и их применение в языковых моделях.

Возможности использования языковых моделей для анализа культурных текстов и творческого контента. Примеры применения языковых моделей в культуре. Автоматический перевод и синтез речи. Технологии генерации текста и их роль в культурном творчестве. Анализ семантики и контекста в культурных текстах с помощью языковых моделей. Возможности применения ИИ для редактирования и улучшения текстового контента.

Преимущества и ограничения использования языковых моделей в культурном контексте. Проблемы и вызовы в развитии языковых моделей для культурного творчества. Влияние ИИ на процессы создания, распространения и сохранения культурного наследия. Перспективы развития и использования языковых моделей в сфере культуры.

##### ***Тема 2. Этика использования искусственного интеллекта и его влияние на сферу культуры***

Этические нормы и научно-технический прогресс. Искусственный интеллект: правовые аспекты. Этические принципы применения искусственного интеллекта. Этика в правовом регулировании искусственного интеллекта. Правовые принципы использования искусственного интеллекта.

Этические аспекты использования ИИ в создании культурных продуктов и их распространении. Защита интеллектуальной собственности и авторских прав в контексте использования ИИ в культуре. Возможные последствия автоматизации культурных процессов с помощью ИИ на социальном и культурном уровнях.

Ответственность за содержание и контекст культурных продуктов, созданных с использованием ИИ. Публичный диалог и участие общества в определении этических стандартов для использования ИИ в культуре. Этические дилеммы в создании и использовании генеративного искусства

с помощью ИИ. Профессиональные стандарты и кодексы этики для специалистов, работающих с ИИ в культуре.

### ***Тема 3. Промпт-инжиниринг: понятие и основные принципы***

Основные понятия, концепции и подходы в промпт-инжиниринге. Интеграция промпт-инжиниринга в существующие рабочие процессы в культурной индустрии. Примеры применения промпт-инжиниринга в различных областях культуры. Перспективы использования промпт-инжиниринга для инноваций в культурной сфере.

Роль промпт-инжиниринга в стимулировании творческого процесса в культурной сфере. Создание эффективных промптов для конкретных культурных задач и целей, различных жанров и форматов культурного контента. Обучение и развитие специалистов в области промпт-инжиниринга для культурных целей.

### ***Тема 4. Разработка промптов для сферы культуры***

Промпт-инжиниринг в сфере визуального искусства: создание промптов для генерации графических изображений и видеоматериалов. Разработка промптов для создания литературных произведений с использованием ИИ. Использование промптов для разработки сценариев и сценических произведений. Промпт-инжиниринг для создания музыкальных композиций и звукового дизайна. Разработка промптов для создания культурно-исторических реконструкций и виртуальных туров. Использование промптов для создания интерактивных инсталляций и выставочных проектов. Промпт-инжиниринг как инструмент для создания интерактивных искусственных персонажей и игровых сценариев. Промпты для создания текстов культурных мероприятий, рекламных материалов и программных буклетов.

Промпт-инжиниринг в маркетинге сферы культуры: разработка промптов в соответствии с профилем целевой аудитории. Автоматизация процесса разработки промптов.

### ***Тема 5. Информационные продукты на основе искусственного интеллекта для решения практических задач в сфере культуры***

Понятие сервисов на основе ИИ и их роль в культурной сфере. Применение сервисов на основе ИИ для анализа культурного контента и выявления его ключевых характеристик.

ИИ-ассистенты для культурных туров и экскурсий учреждений сферы культуры. Применение инструментов ИИ для цифровой каталогизации культурного наследия с помощью ИИ. ИИ-инструменты для виртуального обучения и обмена культурным опытом.

Перспективы развития и интеграции программируемых сервисов на основе ИИ в различные аспекты социокультурной деятельности.

## 5.2 Учебно-методическая карта учебной дисциплины

| Разделы и темы                                                                                                             | Количество аудиторных часов |        |                      |                      | Количество часов УСП | Форма контроля |
|----------------------------------------------------------------------------------------------------------------------------|-----------------------------|--------|----------------------|----------------------|----------------------|----------------|
|                                                                                                                            | всего                       | лекции | лабораторные занятия | практические занятия |                      |                |
| <b>Тема 1.</b> Понятие ИИ и языковых моделей                                                                               | 8                           | 8      |                      |                      |                      |                |
| <b>Тема 2.</b> Этика использования ИИ и его влияние на сферу культуры                                                      | 4                           |        |                      | 4                    |                      |                |
| <b>Тема 3.</b> Промт-инжиниринг: понятие и основные принципы                                                               | 8                           | 2      |                      | 6                    |                      | отчёт          |
| <b>Тема 4.</b> Разработка промтов для сферы культуры                                                                       | 10                          | 2      |                      | 8                    |                      | проект         |
| <b>Тема 5.</b> Информационные продукты на основе искусственного интеллекта для решения практических задач в сфере культуры | 14                          |        | 14                   |                      |                      | проект         |
| <b>Всего аудиторных</b>                                                                                                    | 44                          | 12     | 14                   | 18                   | 46                   | зачет          |
| <b>Всего</b>                                                                                                               | 90                          |        |                      |                      |                      |                |

### 5.3 Список основной литературы

1. Кревецкий, А. В. Основы технологий искусственного интеллекта [Электронный ресурс]: учеб. пособие : [16+] / А. В. Кревецкий, Ю. А. Ипатов, Н. И. Роженцова ; под общ. ред. А. В. Кревецкого ; Поволжский государственный технологический университет. – Йошкар-Ола : Поволжский государственный технологический университет, 2023. – 272 с. : ил., табл., схем. – Режим доступа: <https://biblioclub.ru/index.php?page=book&id=714624>.
2. Фролов, С. А. Искусственный интеллект и архитектура сознания / Сергей Фролов ; [отв. ред. А. Сидорович]. - Москва : Acta Diurna, 2023. – С. 25-191. табл.

### 5.4 Список дополнительной литературы

1. Андрейчиков, А. В. Интеллектуальные информационные системы и методы искусственного интеллекта [Электронный ресурс] : учебник / А.В. Андрейчиков, О.Н. Андрейчикова. — Москва : ИНФРА-М, 2024. — 530 с. [Режим доступа : <https://znanium.ru/catalog/product/2132501>]
2. Баяк, Д. А. Правовые и этические проблемы искусственного интеллекта [Электронный ресурс] : учебник для магистратуры / Д. А. Баяк, А. В. Попова. - Москва : Прометей, 2022. - 300 с. Режим доступа : <https://znanium.ru/catalog/product/2124861>
3. Бутл, Р. Искусственный интеллект и экономика : Работа, богатство и благополучие в эпоху мыслящих машин [Электронный ресурс] : научно-популярное издание / Р. Бутл. - Москва : Альпина ПРО, 2023. - 424 с. [Режим доступа : <https://znanium.ru/catalog/product/2141006>]
4. Дейвенпорт, Т. Внедрение искусственного интеллекта в бизнес-практику: преимущества и сложности [Электронный ресурс] : практическое руководство / Т. Дейвенпорт, З. Мамедьяров. - Москва : Альпина Паблишер, 2021. - 316 с. [Режим доступа : <https://znanium.ru/catalog/product/1841898>]
5. Киссинджер, Г. Искусственный разум и новая эра человечества [Электронный ресурс] : научно-популярное издание / Г. Киссинджер, Э. Шмидт, Д. Хаттенлокер. - Москва : Альпина ПРО, 2022. - 200 с. [Режим доступа : <https://znanium.com/catalog/product/1905830>]
6. Маркус, Г. Искусственный интеллект: Перезагрузка. Как создать машинный разум, которому действительно можно доверять [Электронный ресурс] : практическое руководство / Г. Маркус, Э. Дэвис. -

Москва : Альпина ПРО, 2021. - 300 с. [Режим доступа : <https://znanium.com/catalog/product/1905852>]